



**TUGAS AKHIR - KS141501**

## **OTOMASI REGISTRASI PESERTA ISICO BERBASIS PROSES DENGAN BONITA BPM**

### ***AUTOMATING ISICO'S REGISTRATION BASED ON PROCESS WITH BONITA BPM***

**DELINA RAHAYU EFFENDI**  
**NRP 5213 100 106**

**Dosen Pembimbing**  
**Mahendrawathi ER, S.T., M.Sc., Ph.D**  
**Amna Shifia Nisafani, S.Kom., M.Sc.**

**JURUSAN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2017**

**TUGAS AKHIR - KS141501**

# **OTOMASI REGISTRASI PESERTA ISICO BERBASIS PROSES DENGAN BONITA BPM**

**DELINA RAHAYU EFFENDI**

**NRP 5213 100 106**

**Dosen Pembimbing**

**Mahendrawathi ER, S.T., M.Sc., Ph.D**  
**Amna Shifia Nisafani, S.Kom., M.Sc.**

**JURUSAN SISTEM INFORMASI**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2017**

**UNDERGRADUATE THESIS - KS 141501**

***AUTOMATING ISICO'S REGISTRATION BASED  
ON PROCESS WITH BONITA BPM***

**DELINA RAHAYU EFFENDI**

**NRP 5213 100 106**

**Supervisor**

**Mahendrawathi ER, S.T., M.Sc., Ph.D**

**Amna Shifia Nisafani, S.Kom., M.Sc.**

**INFORMATION SYSTEMS DEPARTMENT  
Information Technology Faculty  
Sepuluh Nopember Institut of Technology  
Surabaya 2017**

## LEMBAR PENGESAHAN

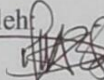
### OTOMASI RESGISTRASI PESERTA ISICO BERBASIS PROSES DENGAN BONITA BPM

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
Pada

Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh



**DELINA RAHAYU EFFENDI**

NRP. 5213100106

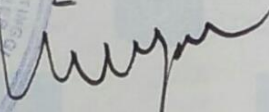
Surabaya, 6 Juli 2017

**KETUA  
JURUSAN SISTEM INFORMASI**



**Dr. Ir. Aris Tjahyanto, M.Kom.**

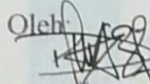
NIP. 196503101991021001



## LEMBAR PERSETUJUAN

### OTOMASI REGISTRASI PESERTA ISICO BERBASIS PROSES DENGAN BONITA BPM TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
Pada  
Jurusan Sistem Informasi  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh: 

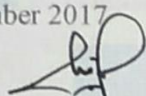
**DELINA RAHAYU EFFENDI**

NRP. 5213100106


Disetujui Tim Penguji: Tanggal Ujian: 31 Mei 2017

Periode Wisuda: September 2017

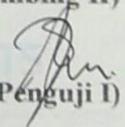
**Mahendrawathi ER, S.T., M.Sc., Ph.D**

  
(Pembimbing I)

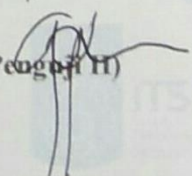
**Amna Shifia Nisafani, S.Kom., M.Sc.**

 an. nissen asrul sari  
(Pembimbing II)

**Erma Suryani, S.T., M.T., Ph.D..**

  
(Penguji I)

**Arif Wibisono, S.Kom., M.Sc.**

  
(Penguji II)



# **OTOMASI REGISTRASI PESERTA ISICO BERBASIS PROSES DENGAN BONITA BPM**

**Nama Mahasiswa : Delina Rahayu Effendi**

**NRP : 5213100106**

**Jurusan : Sistem Informasi FTIF-ITS**

**Pembimbing I : Mahendrawathi Er., S.T., M.Sc., Ph.D**

**Pembimbing II : Amna Shifia Nisafani, S.Kom., M.Sc.**

## **ABSTRAK**

*The Information Systems International Conference (ISICO) merupakan salah satu konferensi internasional yang diselenggarakan oleh Jurusan Sistem Informasi ITS. Peserta ISICO dapat mendaftarkan diri menjadi pemakalah atau non pemakalah. Proses pendaftaran peserta selama ini dilakukan tanpa menggunakan sebuah sistem, yaitu dengan mengirimkan email bukti pendaftaran dan pembayaran. Hal ini membuat segala aktivitas tidak terintegrasi dengan baik dan berdampak ketika pelaksanaan ISICO berlangsung, dimana bagian registrasi kesulitan dalam menentukan apakah seorang peserta telah membayar lunas atau belum.*

*Mengingat proses tersebut terdiri dari berbagai aktivitas yang harus dikerjakan secara bertahap, penulis mengusulkan untuk membuat sistem pendaftaran dengan pendekatan business process management (BPM). BPM ini digunakan agar setiap aktivitas yang terjadi dalam proses pendaftaran telah dilakukan peserta sehingga penyelenggara dapat memantau aktivitas peserta. Selain itu, dengan menggunakan BPM, sistem dapat diperbarui dengan mudah bila terjadi perubahan pada prosesnya. Untuk mengotomasi proses ini, penulis menggunakan Bonita BPM yang merupakan software untuk*

*mengotomasi proses menjadi sistem dan tidak berbayar. Bonita BPM juga dirancang untuk menyelesaikan masalah dengan pendekatan proses dan mudah dalam penggunaan melalui fitur-fitur yang ada. Pemodelan dalam software tersebut telah menggunakan business process modeling notation (BPMN) dan untuk mendesainnya cukup mudah, yaitu dengan drag & drop.*

*Hasil dari tugas akhir ini adalah aplikasi berbasis proses yang dapat menangani proses pendaftaran peserta ISICO dimulai dari mengisi formulir pendaftaran, pembayaran, dan formulir pembenaran nama bila diperlukan. Proses pendaftaran menjadi terotomasi dan terintegrasi dalam sebuah database.*

***Kata kunci:*** *registration systems, ISICO, business process management, business process automation, Bonita BPM*



**PROCESS BASED IMPLEMENTATION OF  
CONFERENCE PARTICIPANTS REGISTRATION  
SYSTEMS WITH BONITA BPM  
(CASE STUDY: ISICO)**

**Nama Mahasiswa : Delina Rahayu Effendi**

**NRP : 5213100106**

**Jurusan : Sistem Informasi FTIF-ITS**

**Pembimbing I : Mahendrawathi Er., S.T., M.Sc., Ph.D**

**Pembimbing II : Amna Shifia Nisafani, S.Kom., M.Sc.**

**ABSTRACT**

*The Information Systems International Conference (ISICO) is one of international conference that held by Department of Information Systems in ITS. ISICO participants can register themselves as a speaker or non-speaker. The registration process is done without using certain system, just by sending the proof of registration and payment by e-mail. This causes all of the activities are not integrated well and affect the execution of ISICO, which is the registration unit is having trouble in determining participant's status of payment.*

*Considering the process itself, consists of various activities that have to be done gradually, the writer proposes to build registration system with business process management (BPM) approach. BPM is used in order to make every activities included in the registration process can be monitored by the event organizer. In addition, the system can be renewed easily if there is any change in the occurring process. To automate this process, writer uses Bonita BPM because it is free and automates process to system. Bonita BPM also designed for*

*solving a process-based problem and easy to use with the features. The modeling in the software uses business process modeling notation (BPMN) and easy to design, only by drag and drop.*

*The result of this thesis is a process based system that can handle the registration process of ISICO, starting from filling out the registration form, payment, and name justification form if necessary. The registration process becomes automated and integrated in a database.*

**Keywords:** *registration systems, ISICO, business process management, business process automation, Bonita BPM*

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT atas limpahan karunia, rahmat, serta barokah yang diberikan-Nya selama ini sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **Otomasi Registrasi Peserta ISICO Berbasis Proses dengan Bonita BPM**. Terima kasih kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, bantuan baik materi maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus, penulis akan menyampaikan terima kasih banyak kepada:

1. Allah SWT.
2. Papa (Dody), Mama (Sutiah), kakak (Eko, Erik, dan Ningsih), dan adik (Felix, Firzal, dan Rasyid) penulis yang tiada henti memberikan dukungan dan semangat.
3. Ibu Mahendrawathi ER., S.T., M.Sc., Ph.D. selaku dosen wali dan dosen pembimbing pertama penulis yang senantiasa meluangkan waktu, memberikan ilmu serta petunjuk, dan memotivasi untuk kelancaran tugas akhir ini.
4. Ibu Amna Shifia Nisafani, S.Kom., M.Sc., selaku dosen pembimbing kedua dan sebagai narasumber yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir.
5. Ibu Erma Suryani, S.T., M.Sc., Ph.D., dan Bapak Arif Wibisono, S.Kom., M.Sc., selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.
6. Bapak Faizal Mahananto, S.Kom., M.Eng., sebagai narasumber dan penguji sistem, yang telah meluangkan waktu serta memberikan ilmu dan petunjuk serta motivasi untuk kelancaran tugas akhir.
7. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
8. Chandra Surya Wicaksono selaku sahabat sepenanggungan namun tidak senasib dalam pengerjaan tugas akhir ini.

9. Stezar Priansya, M. Zuhri, Marina Safitri, Oky Putra selaku sahabat yang telah memberikan ilmu dan pencerahan terkait perancangan sistem dalam tugas akhir ini.
10. Marina Safitri, Shania Olivia, Pramita Lucianna, Provani Winda, Stezar Priansya, Chandra Surya, Ikhwan Aziz, M. Fahmi, Nadya Chandra, Caesar Gilang, Rani Oktavia, Bintang Setyawan yang telah mendukung dan menemani penulis dari masa mahasiswa baru hingga tugas akhir ini dapat diselesaikan.
11. Rovia, Pega, Aini, Rosa, Nadia, Tri Indah, Adinda, dan Vancha selaku sahabat penulis semasa SMP, SMA, (hingga kini) yang memberikan dukungan serta doa.
12. Mas Brantas selaku orang terdekat penulis yang selalu memberikan dukungan, hiburan, dan doa kepada penulis.
13. Rekan-rekan ITS TV dan BELTRANIS yang telah memberikan banyak kenangan manis dan pahit semasa kuliah.
14. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah turut serta menyukseskan penulis dalam menyelesaikan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 31 Mei 2017

Penulis,

Delina Rahayu Effendi

## DAFTAR ISI

LEMBAR PENGESAHAN.... **Error! Bookmark not defined.**

LEMBAR PERSETUJUAN... **Error! Bookmark not defined.**

ABSTRAK .....v

KATA PENGANTAR .....ix

DAFTAR ISI .....xi

DAFTAR GAMBAR .....xvi

DAFTAR TABEL .....xxiii

BAB I PENDAHULUAN ..... 1

1.1. Latar belakang masalah..... 1

1.2. Rumusan masalah..... 5

1.3. Batasan permasalahan ..... 6

1.4. Tujuan ..... 6

1.5 Manfaat ..... 7

1.5.1 Bagi penulis..... 7

1.5.2 Bagi ISICO..... 7

1.6 Relevansi ..... 7

BAB II TINJAUAN PUSTAKA..... 9

2.1 Penelitian sebelumnya..... 9

2.2 Dasar teori ..... 10

2.2.1 Konferensi ..... 10

2.2.2 Business process management ..... 12

2.2.3 Business process modelling notation ..... 15

2.2.4 Bonita BPM..... 19

2.2.5	Implementasi Sistem.....	21
2.2.6	Pengujian Sistem .....	21
<b>BAB III METODOLOGI .....</b>		<b>23</b>
3.1	Tahapan Pelaksanaan Tugas Akhir.....	25
3.1.1.	Analisis permasalahan .....	25
3.1.2.	Studi literatur .....	25
3.1.3.	Implementasi .....	25
3.1.4.	Pemantauan dan pengontrolan.....	26
3.1.5.	Penyusunan tugas akhir .....	27
<b>BAB IV IMPLEMENTASI .....</b>		<b>29</b>
4.1.	Identifikasi proses as-is ISICO .....	29
4.2.	Pemodelan proses as-is ISICO .....	40
4.3.	Identifikasi proses to-be ISICO .....	44
4.4.	Pemodelan proses to-be ISICO.....	48
4.5.	Sistem Eksternal .....	50
4.6.	Perancangan Formulir untuk User Acceptance Testing	51
4.6.1	Scope user acceptance testing.....	51
4.6.2	Fitur yang diuji coba.....	51
4.6.3	Responsibilities.....	52
4.6.4	Test scripts .....	53
4.6.5	Checklist item .....	62
<b>BAB V IMPLEMENTASI .....</b>		<b>67</b>
5.1.	Penentuan Aktor .....	67
5.2.	Pembuatan Proses Model.....	69

5.2.1.	Menambahkan pool .....	70
5.2.2.	Menambahkan lane .....	70
5.2.3.	Menambahkan start events .....	71
5.2.4.	Menambahkan task dan flow .....	71
5.2.5.	Menambahkan gateway .....	72
5.3.	Pembuatan Variabel .....	72
5.4.	Pembuatan Form .....	74
5.4.1.	Formulir Peserta .....	74
5.4.2.	Formulir Bagian Keuangan .....	94
5.5.	Konfigurasi Fungsi Gateway .....	98
5.6.	Konfigurasi Fungsi Loop .....	100
5.7.	Konfigurasi Connector .....	101
5.7.1.	Connector Email .....	102
5.7.2.	Connector MySQL .....	105
5.8.	Publish Organization .....	108
5.9.	Deklarasi Organisasi .....	109
5.9.1.	Konfigurasi Actor Mapping .....	109
5.9.2.	Konfigurasi Autentikasi .....	109
5.10.	Sistem Eksternal .....	110
5.10.1.	Implementasi Fitur Pendaftaran Akun Bonita	110
5.10.2.	Implementasi Fitur Pembuatan Invoice .....	113
5.10.3.	Implementasi Fitur Pembuatan Payment Receipt	116
5.10.4.	Implementasi Fitur Pembuatan Qrcode .....	119

5.10.5.	Implementasi Fitur Informasi Dashboard.....	121
BAB VI HASIL DAN PEMBAHASAN.....		123
6.1.	Mendaftarkan Akun Bonita BPM.....	123
6.2.	Memilih Mendaftarkan Paper atau Tidak.....	125
6.3.	Menambahkan Paper .....	125
6.4.	Memilih presenter.....	125
6.5.	Menambahkan Paper Tambahan .....	126
6.6.	Memilih Menambahkan Partisipan atau Tidak.....	127
6.7.	Menambahkan Partisipan .....	127
6.8.	Memilih Menambahkan Kebutuhan Lain atau Tidak 128	
6.9.	Memesan Prosiding .....	129
6.10.	Mereservasi Hotel.....	129
6.11.	Membuat Invoice.....	131
6.12.	Mengirimkan Invoice .....	133
6.13.	Mengunggah Bukti Pembayaran .....	133
6.14.	Meverifikasi Bukti Pembayaran .....	135
6.15.	Membuat Qrcode dan Bukti Penerimaan Pembayaran .....	136
6.16.	Mengirim Qrcode dan Bukti Penerimaan Pembayaran .....	139
6.17.	Konfirmasi Kehadiran .....	139
6.18.	Konfirmasi Kebenaran Nama .....	140
6.19.	Mengisi Formulir Pembetulan Nama.....	140
BAB VII KESIMPULAN DAN SARAN .....		143



7.1. Kesimpulan .....	143
7.2. Saran.....	144
DAFTAR PUSTAKA .....	145
Lampiran A. Hasil Wawancara Proses as-is ISICO .....	149
Lampiran B. Diagram Proses (as-is) .....	151
Lampiran C. Hasil Wawancara Proses to-be ISICO .....	159
Lampiran D. Diskusi Proses yang Berjalan pada Aplikasi ..	161
Lampiran E. Diagram Proses (to-be).....	167
Lampiran F. Tampilan Hasil Implementasi.....	171
Lampiran G. Hasil User Acceptance Testing .....	189
Biodata Penulis.....	194

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 1.1 Kerangka kerja riset laborator 1 .....	8
Gambar 2.1 Siklus business process management .....	13
Gambar 2.2 Pool.....	18
Gambar 2.3 Lane .....	18
Gambar 3.1 Diagram metodologi bagian 1 .....	23
Gambar 3.2 Diagram metodologi bagian 2 .....	24
Gambar 5.1 Manage Organization .....	67
Gambar 5.2 Manage group.....	68
Gambar 5.3 Manage role.....	68
Gambar 5.4 Setting user .....	69
Gambar 5.5 Manage user .....	69
Gambar 5.6 Menambahkan pool .....	70
Gambar 5.7 Menambahkan lane.....	70
Gambar 5.8 Menambahkan start events .....	71
Gambar 5.9 Menambahkan task dan flow.....	71
Gambar 5.10 Menambahkan gateway .....	72
Gambar 5.11 Menu Data .....	73
Gambar 5.12 Menambahkan variabel proses .....	73
Gambar 5.13 Variabel yang digunakan .....	74
Gambar 5.14 Tampilan add form .....	75
Gambar 5.15 Menentukan variabel yang digunakan .....	75
Gambar 5.16 Formulir memilih mendaftarkan paper atau tidak .....	76
Gambar 5.17 Script mengambil username pada Bonita .....	76
Gambar 5.18 Konfigurasi penyimpanan pada widget .....	77
Gambar 5.19 Konfigurasi HTML aktivitas memilih paper atau tidak .....	77
Gambar 5.20 Konfigurasi radiobutton .....	78
Gambar 5.21 Tampilan formulir memilih menambahkan paper atau tidak .....	78
Gambar 5.22 Form menambahkan paper .....	79
Gambar 5.23 Tampilan formulir add paper.....	79
Gambar 5.24 Formulir memilih presenter.....	80

Gambar 5.25 Tampilan formulir memilih presenter .....	81
Gambar 5.26 Formulir menambahkan paper tambahan .....	81
Gambar 5.27 Tampilan formulir menambahkan paper tambahan.....	82
Gambar 5.28 Formulir memilih menambah partisipan atau tidak .....	83
Gambar 5.29 Tampilan formulir memilih menambah partisipan atau tidak.....	83
Gambar 5.30 Formulir menambahkan partisipan .....	84
Gambar 5.31 Tampilan formulir menambahkan partisipan....	85
Gambar 5.32 Formulir memilih menambah kebutuhan lain atau tidak .....	85
Gambar 5.33 Tampilan formulir memilih menambah kebutuhan lain atau tidak.....	86
Gambar 5.34 Formulir memesan prosiding .....	86
Gambar 5.35 Tampilan formulir memesan prosiding.....	87
Gambar 5.36 Formulir reservasi hotel .....	88
Gambar 5.37 Tampilan mereservasi hotel .....	88
Gambar 5.38 Formulir mengunggah bukti pembayaran.....	89
Gambar 5.39 Konfigurasi file widget .....	90
Gambar 5.40 Tampilan formulir mengunggah bukti pembayaran.....	90
Gambar 5.41 Formulir konfirmasi kehadiran .....	91
Gambar 5.42 Tampilan formulir konfirmasi kehadiran.....	91
Gambar 5.43 Formulir konfirmasi kebenaran nama .....	92
Gambar 5.44 Tampilan formulir kebenaran nama.....	93
Gambar 5.45 Formulir membenaran nama .....	94
Gambar 5.46 Tampilan formulir membenaran nama.....	94
Gambar 5.47 Formulir mengirimkan invoice .....	95
Gambar 5.48 Tampilan formulir mengirimkan invoice.....	96
Gambar 5.49 Formulir verifikasi bukti pembayaran .....	96
Gambar 5.50 Tampilan formulir verifikasi bukti pembayaran .....	97
Gambar 5.51 Formulir mengirim bukti penerimaan pembayaran dan qrcode .....	98

Gambar 5.52 Tampilan formulir mengirim bukti penerimaan pembayaran dan qrcode.....	98
Gambar 5.53 Penggunaan connector gateway bertipe XOR..	99
Gambar 5.54 Konfigurasi default flow pada XOR gateway	100
Gambar 5.55 Konfigurasi penentuan kondisi pada XOR gateway .....	100
Gambar 5.56 Konfigurasi fungsi loop.....	101
Gambar 5.57 Konfigurasi connector .....	102
Gambar 5.58 Memilih connector messaging.....	102
Gambar 5.59 Konfigurasi informasi koneksi email .....	103
Gambar 5.60 Konfigurasi alamat email .....	103
Gambar 5.61 Konfigurasi konten email .....	104
Gambar 5.62 Konfigurasi SMTP email.....	104
Gambar 5.63 Memilih jenis connector database MySQL ....	105
Gambar 5.64 Konfigurasi driver database.....	106
Gambar 5.65 Konfigurasi informasi pengaksesan database.	106
Gambar 5.66 Editor untuk memasukkan query .....	106
Gambar 5.67 Konfigurasi keluaran operasi select database.	107
Gambar 5.68 Script menyimpan data ke variabel proses .....	107
Gambar 5.69 Konfigurasi pengguna default sistem .....	108
Gambar 5.70 Publish organization .....	108
Gambar 5.71 Konfigurasi actor mapping .....	109
Gambar 5.72 Konfigurasi autentikasi pengguna .....	110
Gambar 5.73 Tampilan fitur pendaftaran akun Bonita.....	111
Gambar 5.74 Potongan kode pada controllers fitur pendaftaran akun Bonita .....	111
Gambar 5.75 Potongan kode pada model fitur pendaftaran akun Bonita .....	112
Gambar 5.76 Potongan kode controller penyimpanan data registrasi akun Bonita.....	113
Gambar 5.77 Tampilan fitur pembuatan invoice.....	114
Gambar 5.78 Potongan kode pada controllers fitur pembuatan invoice.....	115
Gambar 5.79 Potongan kode pada model fitur pembuatan invoice.....	116

Gambar 5.80 Tampilan fitur payment receipt.....	116
Gambar 5.81 Potongan kode pada controllers payment receipt .....	118
Gambar 5.82 Tampilan profil presenter pada qrcode .....	119
Gambar 5.83 Potongan syntax untuk membuat aksi link .....	119
Gambar 5.84 Potongan kode pada controllers pembuatan qrcode .....	120
Gambar 5.85 Tampilan dashboard informasi .....	121
Gambar 5.86 Potongan kode pada controllers fitur dashboard .....	122
Gambar 6.1 Mendaftar akun bonita.....	123
Gambar 6.2 Hasil penyimpanan data pendaftaran akun .....	124
Gambar 6.3 Hasil pemberitahuan email akun telah terdaftar .....	124
Gambar 6.4 Hasil pembaruan dashboard.....	124
Gambar 6.5 Hasil pembaruan database terkait paper .....	125
Gambar 6.6 Hasil pembaruan database terkait presenter.....	126
Gambar 6.7 Hasil pembaruan author registrations pada dashboard.....	126
Gambar 6.8 Hasil pembaruan database terkait penambahan additional paper .....	127
Gambar 6.9 Hasil pembaruan database terkait pendaftaran partisipan .....	128
Gambar 6.10 Hasil pembaruan jumlah partisipan pada dashboard.....	128
Gambar 6.11 Hasil pembaruan database terkait pemesanan prosiding.....	129
Gambar 6.12 Hasil pembaruan database terkait data reservasi hotel .....	130
Gambar 6.13 Hasil pemberitahuan email kepada bagian keuangan.....	130
Gambar 6.14 Hasil pemberitahuan email kepada peserta.....	131
Gambar 6.15 Hasil pembaruan database terkait adanya invoice baru .....	131
Gambar 6.16 Daftar yang harus dibuatkan invoice .....	131

Gambar 6.17 Hasil pembuatan invoice .....	132
Gambar 6.18 Preview invoice sebelum diunduh.....	132
Gambar 6.19 Hasil pengiriman invoice melalui email.....	133
Gambar 6.20 Tampilan validator pada pengunggahan bukti pembayaran .....	134
Gambar 6.21 Hasil pengiriman bukti pembayaran.....	134
Gambar 6.22 Hasil pemberitahuan terkait verifikasi pembayaran .....	135
Gambar 6.23 Hasil pembaruan database pada tabel invoice	136
Gambar 6.24 Hasil pembaruan database pada tabel author..	136
Gambar 6.25 Hasil pembaruan database pada tabel partisipan .....	136
Gambar 6.26 Hasil pembaruan dashboard terkait pelunasan invoice .....	136
Gambar 6.27 Daftar yang harus dibuatkan qrcode.....	137
Gambar 6.28 Tampilan saat akan membuat qrcode .....	137
Gambar 6.29 Hasil pengunduhan qrcode .....	138
Gambar 6.30 Hasil qrcode.....	138
Gambar 6.31 Preview receipt sebelum diunduh.....	138
Gambar 6.32 Hasil pemberitahuan dan pengiriman qrcode dan bukti penerimaan pembayaran .....	139
Gambar 6.33 Hasil pemberitahuan terkait ketidakhadiran peserta .....	140
Gambar 6.34 Hasil pemberitahuan selesainya proses pendaftaran.....	140
Gambar 6.35 Hasil penyimpanan data pembenaran nama ...	141
Gambar B.1 Diagram proses as-is non pemakalah bagian 1	151
Gambar B.2 Diagram proses as-is non pemakalah bagian 2	152
Gambar B.3 Diagram proses as-is non pemakalah bagian 3	153
Gambar B.4 Diagram proses as-is pemakalah bagian 1 .....	154
Gambar B.5 Diagram proses as-is pemakalah bagian 2 .....	155
Gambar B.6 Diagram proses as-is pemakalah bagian 3 .....	156
Gambar B.7 Diagram proses as-is pemakalah bagian 4 .....	157
Gambar E.1 Diagram proses to-be bagian 1.....	167
Gambar E.2 Diagram proses to-be bagian 2.....	168

Gambar E.3 Diagram proses to-be bagian 3.....169



## DAFTAR TABEL

Tabel 2.1 Flow objects .....	15
Tabel 2.2 Connecting objects .....	16
Tabel 2.3 Artifacts.....	18
Tabel 4.1 Identifikasi permasalahan pada non pemakalah/partisipan.....	30
Tabel 4.2 Identifikasi permasalahan pada pemakalah/author	32
Tabel 4.3 Kesimpulan permasalahan pada proses pendaftaran ISICO .....	40
Tabel 4.4 Pemetaan permasalahan dan solusi proses pendaftaran ISICO .....	44
Tabel 4.5 Aktivitas pada proses to-be ISICO .....	46
Tabel 4.6 Peran dalam UAT.....	52
Tabel 4.7 Script 1 .....	53
Tabel 4.8 Script 2 .....	57
Tabel 4.9 Script 3 .....	59
Tabel 4.10 Checklist item keberhasilan aktivitas pada proses	62
Tabel 4.11 Checklist item keberhasilan notifikasi email .....	63
Tabel 4.12 Checklist item keberhasilan fungsional database.	64
Tabel 4.13 Checklist item fungsional sistem eksternal .....	66
Tabel A.1 Informasi wawancara proses as-is .....	149
Tabel A.2 Hasil wawancara dengan narasumber .....	149
Tabel C.1 Informasi wawancara proses to-be .....	159
Tabel C.2 Hasil wawancara proses to-be .....	159
Tabel D.1 Informasi diskusi proses yang berjalan pada aplikasi .....	161
Tabel D.2 Hasil diskusi proses yang berjalan pada aplikasi	161

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

Dalam bab ini dijelaskan gambaran umum mengenai tugas akhir yang diangkat meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir dan relevansi atau manfaat kegiatan tugas akhir. Selain itu dijelaskan pula mengenai sistematika penulisan untuk memudahkan dalam membaca buku tugas akhir ini.

### **1.1. Latar belakang masalah**

Sebagai seorang akademisi, menghadiri sebuah konferensi terkait bidang ilmu yang ditekuni adalah hal yang sangat penting. Hal ini dikarenakan dengan mengikuti konferensi, baik peserta maupun pemakalah dapat saling berbagi pengalaman, hasil penelitian, maupun pengetahuan dalam bidangnya, tak terkecuali pada bidang teknologi informasi yang terus berkembang pesat. Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember (ITS) Surabaya merupakan salah satu jurusan yang menekuni bidang teknologi informasi di Indonesia. Untuk memberikan sumbangsih terhadap dunia teknologi informasi, Jurusan Sistem Informasi menginisiasi sebuah konferensi bertaraf internasional.

*The Information Systems International Conference (ISICO)* adalah sebuah konferensi internasional yang didedikasikan untuk kemajuan teori dan praktik dalam Sistem Informasi. Konferensi ini diadakan setiap dua tahun sekali [1] dan sudah ada sejak tahun 2011 dengan peserta yang berasal dari dalam maupun luar negeri. Untuk dapat berpartisipasi sebagai non pemakalah atau pemakalah, peserta harus mengikuti prosedur yang berlaku. Salah satu prosedur untuk dapat mengikuti ISICO yaitu melakukan proses pendaftaran yang saat ini dilakukan melalui pengiriman *email* yang tertuju pada *email*

penyelenggara. Proses pendaftaran ini meliputi tiga proses, yaitu pendaftaran sebelum konferensi, pendaftaran saat konferensi dilaksanakan, dan khusus untuk pemakalah terdapat pendaftaran saat akan melakukan presentasi makalahnya. Proses pendaftaran sebelum konferensi adalah sebagai berikut:

1. Untuk peserta non pemakalah, proses yang dilakukan adalah melakukan pendaftaran dan mengirim bukti pembayaran melalui *email* tertuju pada *email* penyelenggara ISICO. Kemudian pendaftaran tersebut direkap menggunakan Google Sheet oleh bagian registrasi. Dari bagian registrasi akan melakukan verifikasi bukti pembayaran kepada bagian keuangan. Bila bukti telah terverifikasi, maka bagian keuangan akan mengeluarkan bukti penerimaan sebagai tanda bukti pelunasan biaya pendaftaran. Bukti penerimaan tersebut kemudian diberikan kepada bagian registrasi lalu dikirimkan kepada *email* peserta. Selain itu, bagian registrasi di sini juga melakukan verifikasi kehadiran peserta serta nama lengkap untuk nantinya dituliskan pada sertifikat peserta. Hal ini juga dilakukan melalui *email*. Sedangkan bila bukti pembayaran tadi tidak terverifikasi, maka bagian keuangan akan memberikan notifikasi kepada bagian registrasi untuk kemudian diteruskan kepada peserta melalui *email*. Kondisi ini tidak akan terlalu menjadi masalah bila jumlah peserta tidak banyak. Namun bila jumlah peserta banyak maka akan menyulitkan pihak penyelenggara.
2. Untuk proses pendaftaran sebagai pemakalah juga dimulai dengan pengiriman *email* kepada penyelenggara. Namun di sini, pemakalah harus menyertakan juga makalah yang telah melalui proses *camera ready accepted*. Setelah itu, bagian registrasi juga melakukan rekap pendaftaran pemakalah dengan Google Sheet kemudian melakukan verifikasi kepada bagian keuangan untuk bukti pembayaran serta verifikasi kepada bagian *editor* untuk memastikan

jumlah halaman pada pemakalah sesuai dengan ketentuan. Bila terdapat kelebihan halaman, bagian *editor* akan memberikan notifikasi kepada bagian keuangan bahwa terdapat biaya tambahan yang harus dibayar oleh pemakalah. Dari bagian keuangan, akan memberikan notifikasi nominal biaya tambahan yang harus dibayar pemakalah kepada bagian registrasi. Kemudian dari bagian registrasi akan mengirimkan notifikasi tersebut melalui *email*. Akan tetapi, bila jumlah halaman sesuai dengan ketentuan maka *editor* akan memberikan notifikasi kepada bagian keuangan bahwa tidak ada kelebihan halaman serta bila bukti pembayaran telah terverifikasi, maka bagian keuangan akan mengeluarkan bukti penerimaan kepada pemakalah melalui bagian registrasi seperti pada proses non pemakalah.

Permasalahan lain yang juga muncul di proses ini adalah adanya berbagai jenis pengeluaran bukti penerimaan yang diinginkan oleh peserta. Contohnya jika terdapat dua makalah namun dengan penulis pertama yang berbeda meminta sebuah bukti penerimaan menjadi satu. Hal ini sama dengan kondisi sebelumnya, bila jumlah pemakalah yang mendaftar tercatat banyak, akan menyulitkan pihak penyelenggara dikarenakan proses-proses ini tidak dilakukan dalam satu sistem.

Proses pendaftaran selanjutnya adalah pendaftaran saat konferensi berlangsung. Akibat dari kedua kondisi dari proses pendaftaran non pemakalah dan pemakalah di atas, penyelenggara ISICO kesulitan khususnya dalam hal verifikasi terhadap peserta yang belum melunasi biaya pendaftaran meskipun telah ada rekap pendaftaran. Hal ini dikarenakan rekap pendaftaran tersebut masih dilakukan secara manual di masing-masing bagian, yaitu registrasi, keuangan, dan/atau *editor* (untuk pemakalah) dan tidak terintegrasi dalam sebuah

*database* sehingga butuh waktu yang cukup lama untuk melakukan verifikasi pembayaran pada banyak peserta.

Dari kedua proses pendaftaran tersebut dapat disimpulkan bahwa permasalahan utama yang terjadi pada proses pendaftaran ISICO adalah tidak terintegrasinya proses pendaftaran sehingga menyulitkan panitia dalam melakukan verifikasi pelunasan biaya pendaftaran oleh peserta. Permasalahan ini seharusnya dapat diatasi dengan menggunakan sistem informasi. Sistem informasi yang telah digunakan oleh penyelenggara ISICO adalah *Easychair*. Akan tetapi, fungsi dari sistem tersebut hanya untuk memberikan info tentang diterima atau tidaknya makalah namun tidak termasuk proses pembayaran dikarenakan fitur tersebut memerlukan biaya yang tidak sedikit serta nominalnya bergantung pada banyaknya pengguna. Sedangkan, penyelenggara ISICO menginginkan adanya sebuah sistem informasi yang dapat memfasilitasi proses pendaftaran ISICO yang terintegrasi sehingga dapat diketahui status akhir pembayaran masing-masing peserta. Dari sistem tersebut diharapkan dapat membantu proses pendaftaran ulang saat pelaksanaan ISICO berlangsung dan juga dapat menangani pendaftaran ulang untuk pemakalah yang akan melakukan presentasi. Oleh karena itu, sistem yang dibangun nantinya akan dilengkapi dengan fitur *qr code* untuk mempermudah semua proses pendaftaran ISICO.

Dari penjelasan terkait proses pendaftaran ISICO di atas, dapat dilihat bahwa proses tersebut terdiri dari berbagai aktivitas yang saling terkait serta tidak terintegrasi di dalam sebuah sistem. Untuk dapat memastikan semua aktivitas telah dilakukan oleh peserta, penulis mengusulkan membangun sebuah sistem pembayaran dengan pendekatan berbasis proses. Hal ini juga dapat digunakan penyelenggara untuk memantau proses pendaftaran yang dilakukan oleh peserta.

Disiplin ilmu yang membahas pendekatan ini adalah *Business Process Management* (BPM). Keutamaan dari pendekatan ini adalah bila terjadi perubahan proses atau *business process modeling notation* (BPMN) yang mempengaruhi alur pendaftaran ISICO, aplikasi ini dapat diperbarui dengan lebih mudah sehingga dapat disimpulkan bahwa aplikasi ini dapat dilakukan perubahan sewaktu-waktu, mengikuti proses yang diinginkan atau dibutuhkan.

Untuk menyelesaikan tugas akhir ini, penulis menggunakan Bonita BPM dari Bonitasoft. Bonita BPM merupakan perangkat lunak yang dirancang untuk dapat mengotomasi proses menjadi sebuah sistem atau dengan kata lain Bonita BPM ini dapat digunakan untuk menyelesaikan sebuah permasalahan yang menggunakan pendekatan berbasis proses. Penulis menggunakan Bonita BPM untuk menyelesaikan dan membantu proses otomasi pada tugas akhir ini karena perangkat lunak ini memang dirancang khusus untuk menyelesaikan masalah yang menggunakan pendekatan berbasis proses. Perangkat lunak ini juga tidak berbayar dan pemodelan prosesnya sudah berdasarkan pada BPMN. Selain itu, dalam mendesain antarmuka sistem yang dibangun menggunakan Bonita BPM, pengguna dapat melakukannya melalui *drag & drop*.

## **1.2. Rumusan masalah**

Rumusan masalah dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Bagaimana proses bisnis yang dibutuhkan oleh penyelenggara ISICO untuk mengotomasi dan mengintegrasikan proses pendaftaran ISICO?
2. Bagaimana mengimplementasikan proses pendaftaran ISICO pada Bonita BPM?

3. Apakah sistem yang dikembangkan sudah memenuhi kebutuhan penyelenggara ISICO?

### 1.3. Batasan permasalahan

Sesuai dengan deskripsi permasalahan yang telah dijelaskan diatas, adapun batasan permasalahan dari penyelesaian tugas akhir ini adalah sebagai berikut:

1. Analisis proses bisnis saat ini (*as-is*) menjadi proses bisnis yang diharapkan (*to-be*) tidak dilakukan. Pada tugas akhir ini hanya menggambarkan proses *as-is* dan *to-be* sehingga pengerjaan tugas akhir ini difokuskan pada tahap implementasi dan pengujian sistem.
2. Sistem yang dibangun hanya untuk menangani pendaftaran peserta ISICO sehingga proses penerimaan makalah tidak dilakukan dalam sistem ini.
3. Sistem yang dibangun menangani pendaftaran sebelum konferensi berlangsung.
4. Sistem yang dibangun tidak memfasilitasi fitur pembayaran, namun hanya berupa informasi status pelunasan pembayaran peserta.
5. Sistem Bonita yang dibangun dapat menangani satu peserta untuk setiap proses yang dijalankan.
6. Aktivitas pembuatan akun Bonita, *invoice*, *payment receipt*, *qrcode*, dan informasi *dashboard* dibangun pada sistem yang berbeda dari Bonita BPM, yaitu menggunakan *web programming* dengan *framework codeigniter*.

### 1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Mengidentifikasi proses bisnis yang dapat mengotomasi dan mengintegrasikan semua aktivitas pada proses pendaftaran ISICO.



2. Membangun sistem pendaftaran yang menggunakan Bonita BPM.

### **1.5 Manfaat**

Berikut manfaat yang diperoleh, dengan melihat dari dua sudut pandang, yaitu sudut pandang penulis dan pihak penyelenggara ISICO.

#### **1.5.1 Bagi penulis**

1. Memberikan pembelajaran kepada penulis dalam pembuatan sistem pendaftaran melalui pendekatan berbasis proses.
2. Memberikan pembelajaran dalam mendesain sebuah proses yang berjalan dalam pendaftaran konferensi.

#### **1.5.2 Bagi ISICO**

1. Mempermudah proses pendaftaran bagi pihak penyelenggara dikarenakan sistem yang sudah terintegrasi.
2. Mempermudah proses verifikasi pembayaran biaya pendaftaran oleh peserta ISICO yang akhirnya dapat mempersingkat waktu registrasi ulang saat konferensi berlangsung.
3. Mempercepat proses pendaftaran ulang bagi pemakalah yang akan melakukan presentasi.

### **1.6 Relevansi**

Laboratorium Sistem Enterprise (SE) Jurusan Sistem Informasi ITS Surabaya memiliki empat topik utama (dapat dilihat pada Gambar 1.1), yaitu *Customer Relationship Management* (CRM), *Enterprise Resource Planning* (ERP), *Supply Chain Management* (SCM), dan *Business Process Management* (BPM). Tugas akhir yang dikerjakan penulis adalah tentang BPM yang termasuk salah satu topik utama dari

riset laboratorium SE. Mata kuliah yang bersangkutan dengan topik ini adalah Desain dan Manajemen Proses Bisnis (DMPB).



**Gambar 1.1 Kerangka kerja riset laborator 1**

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini berisikan tinjauan pustaka yang akan digunakan dalam penelitian tugas akhir ini, yang mencakup penelitian-penelitian sebelumnya, dasar teori dan metode yang digunakan selama pengerjaan.

#### **2.1 Penelitian sebelumnya**

Terdapat beberapa penelitian yang memiliki topik serupa dengan penelitian ini, yaitu tentang sistem pendaftaran, pembayaran ataupun pengembangan aplikasi dengan pendekatan *process-oriented*.

Penelitian terkait sistem pendaftaran seminar pernah dilakukan dengan judul “Perancangan Sistem Informasi Pendaftaran Seminar Mahasiswa Pascasarjana Institut Pertanian Bogor (IPB)”. Perancangan ini dilakukan karena sistem pendataan seminar mahasiswa pascasarjana di IPB yang masih dilakukan secara manual oleh administrator. Kelemahan terjadi ketika banyaknya mahasiswa pascasarjana yang mendaftar, yaitu lebih dari 20 mahasiswa per harinya dengan sumber daya yang terbatas dan sistem yang masih konvensional. Dengan adanya penelitian ini, mampu memberikan kemudahan dari segi efisien waktu, kemudahan tata kelola administrasi, dan tidak adanya sistem antrian manual. Kelebihan lain yang diberikan yaitu mengurangi kebutuhan kertas *form* pendaftaran dan jadwal sehingga perancangan sistem ini memberikan 75% keuntungan bagi pegawai, mahasiswa, dan perguruan tinggi. Sistem ini diimplementasikan menggunakan bahasa pemrograman PHP dan *database* MySQL. Hasil yang diperoleh dari penelitian ini adalah keakuratan data yang tinggi yaitu mencapai 70-75% berdasarkan hasil kuesioner pemanfaatan pengembangan SIPASMAPAS [2].

Penelitian terkait sistem pembayaran salah satunya adalah tentang “Konsep Otomatisasi Sistem Pembayaran SPP Online untuk Mengurangi Tingkat Keterlambatan” pada SMK Ricardo Auto Machine. Sebelumnya, sekolah ini telah menggunakan sistem terkomputerisasi dengan harapan data yang ada menjadi lebih akurat, cepat, dan efisien dalam menangani pembayaran SPP siswa. Namun kenyataannya tidak demikian. Dengan adanya keterbatasan sistem yang hanya menggunakan Microsoft Excel, sistem terkomputerisasi ini belum memberikan dampak yang optimal pada SMK Ricardo Auto Machine. Sama seperti sebelumnya, sistem ini dilakukan analisis dengan menggunakan *Unified Modeling Language* (UML) yang menghasilkan *use case diagram*, *class diagram*, *sequence diagram*, dan prototipe. Hasil yang diperoleh dengan adanya sistem pembayaran SPP terotomatisasi ini adalah proses pengolahan data akan menjadi lebih cepat, tepat, dan efisien [3].

Penelitian lain dari Insitut Teknologi Sepuluh Nopember adalah terkait dengan pengembangan aplikasi dengan pendekatan *process-oriented*. Pendekatan ini digunakan untuk mengembangkan sebuah sistem reservasi kendaraan pada PT. Pertamina EP. Penelitian ini mengikuti alur pada BPM, yaitu *process analysis*, *process discovery*, *process redesign*, dan *process implementation*. Pengembangan aplikasi dilakukan dengan menggunakan BONITA BPM yang mampu mengotomatisasi proses bisnis yang berjalan saat reservasi kendaraan di PT. Pertamina EP [4].

## **2.2 Dasar teori**

### **2.2.1 Konferensi**

Berdasarkan *Business Dictionary*, konferensi merupakan sebuah pertemuan besar dari individu-individu atau anggota dari satu atau beberapa organisasi untuk mendiskusikan hal-

hal yang menjadi kepentingan bersama. Sedangkan menurut kamus besar bahasa indonesia (KBBI), konferensi adalah rapat atau pertemuan untuk berunding atau bertukar pendapat mengenai suatu masalah yang dihadapi bersama. Dari dua pengertian tersebut, dapat disimpulkan bahwa konferensi merupakan sebuah pertemuan untuk mendiskusikan atau bertukar pendapat mengenai sebuah topik yang menjadi kepentingan bersama. Beberapa keuntungan ketika seseorang menghadiri konferensi adalah [5] [6] 1) Kesempatan menadapatkan ilmu, 2) Mendapatkan koneksi dengan rekan-rekan sesama peserta konferensi, 3) Dapat memposisikan diri sebagai seorang yang ahli, dan 4) Dapat berintegrasi dengan komunitas akademik.

#### 2.2.1.1. *The Information Systems International Conference (ISICO)*

*The Information Systems International Conference (ISICO)* adalah sebuah konferensi internasional yang didedikasikan untuk kemajuan teori dan praktik dalam Sistem Informasi. Konferensi ini diadakan setiap dua tahun sekali [1] dan sudah ada sejak tahun 2011 dengan peserta yang berasal dari dalam maupun luar negeri.

Tahun 2013, ISICO dihadiri oleh sebanyak 340 partisipan dari 9 negara dan pada 2015, konferensi ini berkolaborasi dengan *Procedia Computer Science* dari Elsevier untuk mempublikasikan semua makalah ISICO menjadi jurnal [7]. Keuntungan yang didapatkan dengan mengikuti ISICO antara lain adalah 1) Dapat membuka wawasan tren teknologi informasi mendatang, 2) Dapat memahami strategi bagaimana menyelaraskan teknologi, bisnis, dan sumber daya manusia, 3) Dapat memahami sistem dan peluang kerja sama penelitian dan pendidikan internasional, serta 4) Dapat memperluas jaringan kerja sama pribadi maupun perusahaan/organisasi [8].

Melihat berbagai keuntungan yang akan didapat melalui keikutsertaan dalam konferensi tersebut, tentunya para akademisi maupun seseorang yang menekuni bidang terkait akan tertarik untuk mengikutinya. Mereka dapat berpartisipasi sebagai non pemakalah maupun pemakalah dengan mengikuti prosedur yang berlaku. Prosedur pendaftaran untuk non pemakalah dapat dilihat pada Lampiran B Gambar B.1 sampai dengan Gambar B.3, sedangkan untuk pemakalah dapat dilihat pada Lampiran B Gambar B.4 sampai dengan Gambar B.7.

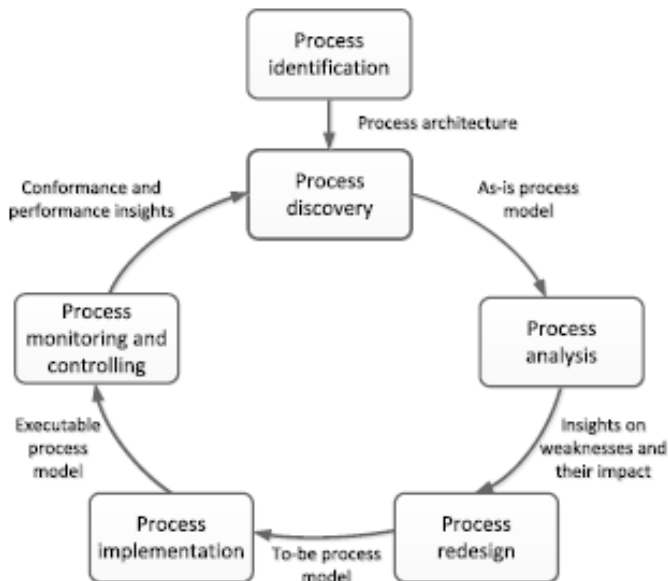
### *2.2.2 Business Process Management*

Selama satu dekade terakhir, proses menjadi sebuah aset penting untuk kehidupan sehari-hari di dalam sebuah organisasi karena kemampuan Business Process Management (BPM) dari organisasi tersebut dapat membantu dalam mencapai tujuannya [9]. BPM sendiri adalah sebuah seni dan ilmu yang melihat bagaimana sebuah pekerjaan dilakukan pada sebuah organisasi untuk memastikan hasil yang konsisten dan mengambil keuntungan dari sebuah perbaikan/perubahan/inovasi. Perbaikan tersebut dapat berupa mengurangi biaya, mengurangi durasi pelaksanaan, dan mengurangi tingkat kesalahan. Hal terpenting yang perlu diketahui adalah, BPM tidak untuk meningkatkan performa aktivitas individu. Tetapi, BPM itu tentang mengelola seluruh rangkaian acara, aktivitas, dan keputusan yang dapat memberikan nilai tambah bagi organisasi [10].

Dalam suatu perusahaan/organisasi, proses bisnis yang dikelola dengan baik akan mampu menumbuhkan peluang. Namun, perusahaan terkadang kurang memahami dan tidak mampu mengontrol proses bisnis yang dimilikinya. Pihak manajemen mungkin telah berhasil membuat prosedur yang ideal untuk menjalankan proses bisnisnya, tapi pada kenyataannya, implementasi di lapangan dapat sangat berbeda dari apa yang telah dirancang. Pada pelaksanaannya, sering

terjadi redundansi, ketidakefisienan, stagnasi, dan berbagai kesalahan lainnya. Dengan BPM atau manajemen proses bisnis, perusahaan dapat mengawasi dan mengontrol seluruh elemen pada proses bisnis [11].

Untuk dapat memanajemen proses bisnis, BPM memiliki sebuah metodologi yang bisa disebut dengan lifecycle (siklus hidup). Siklus BPM dapat dilihat pada Gambar 2.1. Dari gambar tersebut terdapat beberapa proses yang dilakukan, yaitu *process identification*, *process discovery*, *process analysis*, *process redesign*, *process implementation*, dan *process monitoring and controlling* [10].



**Gambar 2.1 Siklus *business process management***

### 1. *Process identification*

Proses identifikasi adalah fase pertama dalam siklus BPM. Di sini, sebuah masalah bisnis akan diangkat/diajukan untuk dianalisa. Hasil yang dikeluarkan dari proses ini adalah

arsitektur proses terbaru atau yang telah diperbarui yang menunjukkan pandangan keseluruhan dari proses di sebuah organisasi dan hubungannya.

## 2. *Process discovery* (dapat disebut juga model proses *as-is*)

Pada fase ini, kondisi saat ini dari setiap proses didokumentasikan. Biasanya dalam bentuk satu atau beberapa model proses *as-is*.

## 3. *Process analysis*

Pada fase ini, isu yang terkait dengan model proses *as-is* diidentifikasi, didokumentasikan, dan bila memungkinkan diukur dengan menggunakan ukuran kinerja. Hasil dari fase ini adalah kumpulan masalah yang terstruktur

## 4. *Process redesign* (dapat disebut dengan *process improvement*)

Tujuan dari fase ini adalah mengidentifikasi perubahan proses yang dapat membantu mengatasi permasalahan yang telah diidentifikasi sebelumnya. Hasil dari fase ini adalah model proses *to-be* yang akan menjadi dasar untuk fase selanjutnya.

## 5. *Process implementation*

Fase ini mencakup dua aspek: manajemen perubahan organisasi dan otomasi proses. Manajemen perubahan organisasi berhubungan dengan kumpulan aktivitas yang dibutuhkan untuk mengubah cara kerja dari seluruh partisipan yang ada di dalam proses. Sedangkan otomasi proses mencakup konfigurasi atau implementasi sebuah sistem untuk mendukung proses *to-be*.

## 6. *Process monitoring and controlling*

Ketika proses *redesign* dijalankan, data terkait akan dikumpulkan dan dianalisis untuk menentukan seberapa baik



proses tersebut berjalan dengan memperhatikan pengukuran dan objektif dari kinerja.


### 2.2.3 *Business process modelling notation*



Standar BPMN dibangun oleh *The Business Process management Initiative* (BPMI). Tujuan utama dari BPMN adalah untuk menyediakan notasi yang siap dipahami oleh semua pelaku bisnis, dari *business analysts* yang membuat *draft* proses awal menuju pengembang teknis yang bertanggung jawab untuk mengimplementasi teknologi yang akan melakukan proses tersebut dan akhirnya sampai kepada *business people* yang akan memantau dan memonitor proses tersebut. BPMN menjelaskan sebuah *Business Process Diagram* (BPD) yang berdasarkan teknik membuat *flowchart* disesuaikan untuk membuat model grafik dari operasi proses bisnis. Terdapat empat kategori dasar dari elemen BPMN, yaitu [9]:

#### 1. *Flow Objects*

Sebuah BPD setidaknya memiliki tiga dari elemen-elemen utama, yang mana merupakan flow objects. Tiga elemen utama tersebut dijelaskan pada Tabel 2.1.

**Tabel 2.1** *Flow objects*

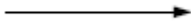
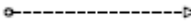
Elemen	Deskripsi	Notasi
<i>Event</i>	Sebuah <i>event</i> ditunjukkan dengan sebuah lingkaran dan menyatakan adanya sesuatu yang terjadi selama perjalanan proses bisnis. Terdapat tiga tipe <i>event</i> berdasarkan pengaruhnya terhadap alur, yaitu <i>Start</i> , <i>Intermediate</i> , dan <i>End</i> .	


Elemen	Deskripsi	Notasi
<i>Activity</i>	Sebuah <i>activity</i> digambarkan dengan kotak bersudut tumpul. Jenis dari elemen ini adalah <i>task</i> dan <i>sub-process</i> . Untuk <i>sub-process</i> nantinya akan dibedakan dengan sebuah simbol tambah (+) kecil di bagian tengah bawah dari elemen ini.	
<i>Gateway</i>	Sebuah <i>Gateway</i> digambarkan dengan bentuk permata dan digunakan untuk mengontrol alur proses atau untuk menentukan pengambilan keputusan, percabangan, atau penggabungan alur.	

## 2. Connecting Objects

Kategori ini berguna untuk menghubungkan *flow objects* untuk membuat struktur kerangka dasar dari proses bisnis. Terdapat tiga penghubung yang dapat dilihat pada Tabel 2.2.

**Tabel 2.2 Connecting objects**

Connector	Deskripsi	Notasi
<i>Sequence Flow</i>	Penghubung ini digambarkan dengan garis lurus dengan anak panah. Penghubung sebagai penunjuk urutan dari proses.	
<i>Message</i>	Penghubung ini digambarkan dengan garis	

<i>Connector</i>	<b>Deskripsi</b>	<b>Notasi</b>
<i>Flow</i>	putus-putus dengan ujung panah terbuka. Penghubung ini digunakan untuk menunjukkan alur dari informasi diantara dua Proses Partisipan yang berbeda. Di dalam BPMN, bila terdapat dua Pool yang berbeda di sebuah diagram menandakan dua Partisipan.	
<i>Association</i>	Penghubung <i>association</i> dilambangkan dengan garis titik-titik dengan ujung panah dari garis. Penghubung ini digunakan untuk menggabungkan data, teks, dan beberapa artifak lainnya dengan flow objects.	

### 3. *Swimlanes*

Beberapa metodologi model proses biasanya menggunakan *swimlanes* sebagai mekanisme untuk mengatur aktivitas-aktivitas menjadi kategori visual yang berbeda agar dapat menggambarkan kemampuan fungsional atau tanggung jawab yang berbeda. Terdapat dua tipe dari BPD *swimlane*, yaitu:

*Pool* menggambarkan partisipan di dalam sebuah proses. *Swimlane* ini juga bertindak sebagai sebuah konten grafis untuk membagi-bagi beberapa aktivitas dari *pool* lain. *Swimlane* ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Pool*

*Lane* merupakan sub-partisi di dalam sebuah *pool* dan akan dilebarkan mengikuti bentuk dari *pool*. Biasanya digunakan untuk mengelola dan mengkategorikan aktivitas yang ada. Bentuk dari *lane* dapat dilihat pada Gambar 2.3.




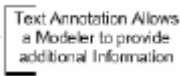
Gambar 2.3 *Lane*

4. *Artifacts*

BPMN didesain untuk memungkinkan pembuat model dan alat-alat pemodel lebih fleksibel dalam mengeksplor notasi. Banyak *artifact* yang dapat ditambahkan ke dalam diagram sesuai dengan konteks dari proses bisnis yang akan dimodelkan. Tabel 2.3 menggambarkan tiga *artifact* dalam versi BPMN.

Tabel 2.3 *Artifacts*

<i>Artifacts</i>	Deskripsi	Notasi
<i>Data Object</i>	Artifact ini adalah sebuah mekanisme untuk menunjukkan bagaimana data dibutuhkan atau diproduksi dari aktivitas. Mereka terhubung dengan	

<i>Artifacts</i>	<b>Deskripsi</b>	<b>Notasi</b>
	aktivitas melalui Associations.	
<i>Group</i>	Sebuah Group ditunjukkan dengan sebuah kotak yang berujung tumpul dengan garis putus-putus. Artifacts ini digunakan untuk tujuan analisis dan dokumentasi, tetapi tidak berpengaruh pada alur sekuens.	
<i>Annotation</i>	Annotation adalah sebuah mekanisme untuk sebuah modeler untuk menyediakan informasi teks tambahan untuk pembaca diagram BPMN.	

#### 2.2.4 Bonita BPM

Bonita BPM adalah sebuah perangkat lunak yang dapat mengatasi alur dari manajemen proses bisnis yang *open-source* dan dibangun sejak 2011. Sejak 2009, pengembangan Bonita didukung oleh perusahaan yang tertarik dengan aktivitas ini, yaitu Bonitasoft [10]. Dengan menggunakan Bonita BPM, para pengembang tidak terbatas dengan fitur-fitur program lain yang kaku sehingga mereka dapat membuat sebuah aplikasi yang benar-benar diinginkan. Keunggulan Bonita BPM antara lain adalah [11]:

1. *Design freely*, yaitu pengguna mendapatkan apa yang diinginkan untuk membangun aplikasi berbasis proses dari alur kerja BPMN menjadi sebuah *state-of-the-art*, dan dapat mendesain antarmuka yang dapat dilakukan dengan *drag & drop*.
2. *Deliver continuously*, yaitu para pengembang tidak perlu lagi melalui siklus pembaruan aplikasi yang panjang dan memakan biaya mahal. Aplikasi yang dibangun dengan Bonita BPM dapat beradaptasi dengan mudah secara *real-time* dengan proses bisnis.
3. *Delight completely*, yaitu para pelaku bisnis mendapatkan aplikasi pribadi untuk melakukan tugas-tugas harian dengan lebih efisien.

Di dalam Bonita BPM terdapat tiga komponen utama, yaitu [10]:

1. Bonita Studio: komponen ini memungkinkan pengguna untuk memodifikasi atau mengubah proses bisnis secara grafik dengan mengikuti standar BPMN. Pengguna juga dapat menghubungkan proses yang ada dengan komponen sistem informasi yang lainnya, seperti *messaging*, *Enterprise Resource Planning (ERP)*, *Enterprise Content Management*, dan *database* dengan tujuan untuk *generate* aplikasi dan bisa diakses dalam bentuk *web*.
2. Bonita BPM Engine: mesin BPM adalah aplikasi JAVA yang mengeksekusi proses yang dibuat dengan Bonita Studio. *Engine API* memungkinkan pengguna untuk berinteraksi secara programatik dengan prosesnya.
3. Bonita Portal: sebuah *portal* yang memungkinkan masing-masing *end-user* untuk memantau dalam *webmail-like interface* untuk semua tugas yang dimiliki. *Portal* ini juga memungkinkan pemilik proses untuk mengelola dan mendapatkan laporan tentang proses yang bersangkutan.

### 2.2.5 Implementasi Sistem

Implementasi sistem dalam siklus BPM diartikan bahwa model *redesign* dari sebuah proses perlu untuk diimplementasikan agar dapat dieksekusi, seperti meletakkan sebuah sistem informasi untuk mencatat dan memantau perjalanan proses. Secara umum, proses implementasi dapat mencakup manajemen perubahan organisasi dan proses otomasi yang telah dijelaskan secara singkat pada sub bab 2.2.2 poin *process implementation*. Aktivitas dalam manajemen perubahan dapat mencakup [12]:

1. Menjelaskan perubahan pada partisipan dengan poin yang dapat dimengerti tentang apa yang diubah dan mengapa hal tersebut dapat menguntungkan perusahaan.
2. Membuat rencana manajemen perubahan sehingga *stakeholder* tahu kapan perubahan akan memberi efek dan rencana transisi apa yang akan dilakukan untuk menyelesaikan masalah selama masa transisi ke proses *to-be*.
3. Melatih pengguna tentang cara kerja yang baru dan memantau perubahan untuk memastikan proses transisi berjalan baik.

### 2.2.6 Pengujian Sistem

Pengujian sistem merupakan sebuah teknik penting untuk menilai kualitas dari produk perangkat lunak. Terdapat dua metodologi umum dalam pengujian sistem [15] [16]: 1) struktural, dimana sebuah teknik yang membutuhkan pemahaman dari kerja internal dari program, 2) fungsional, dimana merupakan sebuah teknik yang mendesain beberapa kasus pengujian berdasarkan informasi dari spesifikasi. Pengujian fungsional sering disebut juga dengan *black-box testing*. Istilah ini menggambarkan bahwa konten (implementasi) dari sebuah *black box* tidak diketahui tetapi

fungsi dari *black box* tersebut benar-benar dapat dipahami dalam lingkup masukan dan keluarannya [17]. Keuntungan utama dari teknik pengujian fungsional adalah penguji tidak perlu memiliki pengetahuan terhadap bahasa pemrograman dan implementasi, karena penguji dan pengembang adalah dua bagian yang terpisah [16]. Selain itu, tes yang dilakukan disesuaikan dengan apa yang diharapkan pada sistem, yang alami dan dimengerti oleh setiap pengguna [18].

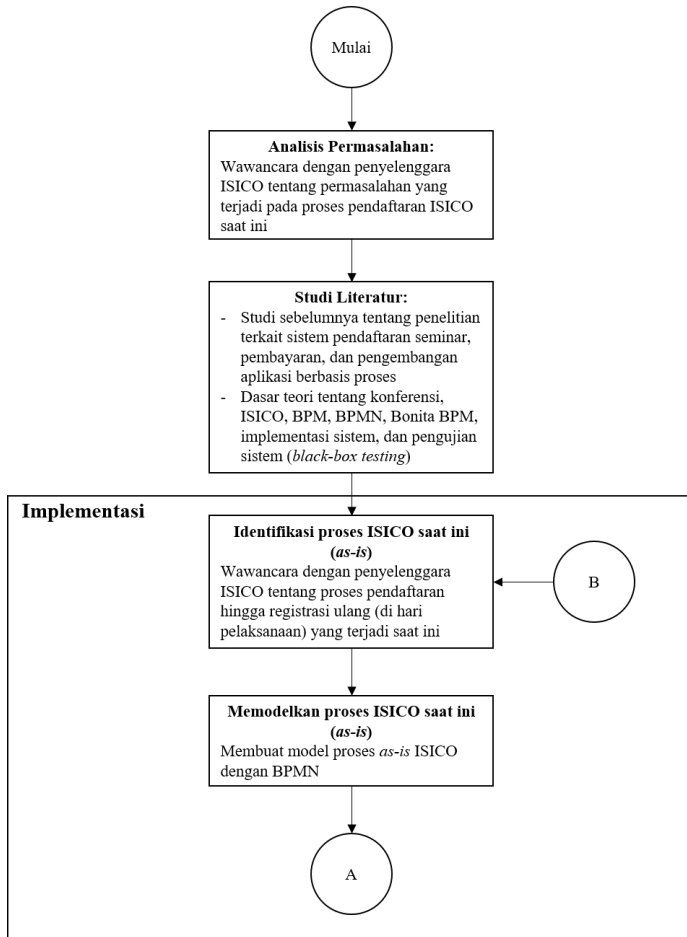
Tujuan dari adanya pengujian sistem adalah untuk melakukan verifikasi dan validasi. Pada tugas akhir ini, penulis akan melakukan validasi sistem sehingga menggunakan teknik *black-box testing*. Penggunaan teknik ini dikarenakan *black-box testing* sering digunakan untuk melakukan validasi. Selain itu, pengguna atau penguji tidak perlu memiliki pengetahuan secara teknis, karena yang diuji adalah fungsional sistem [16] sehingga hasil dari pengujian menggunakan *black-box* adalah untuk mengetahui kesenjangan antara sistem yang telah dibangun dengan spesifikasi yang dibutuhkan oleh pengguna atau secara lebih singkat dapat digunakan untuk mencari eror atau kesalahan dari sistem yang diuji.



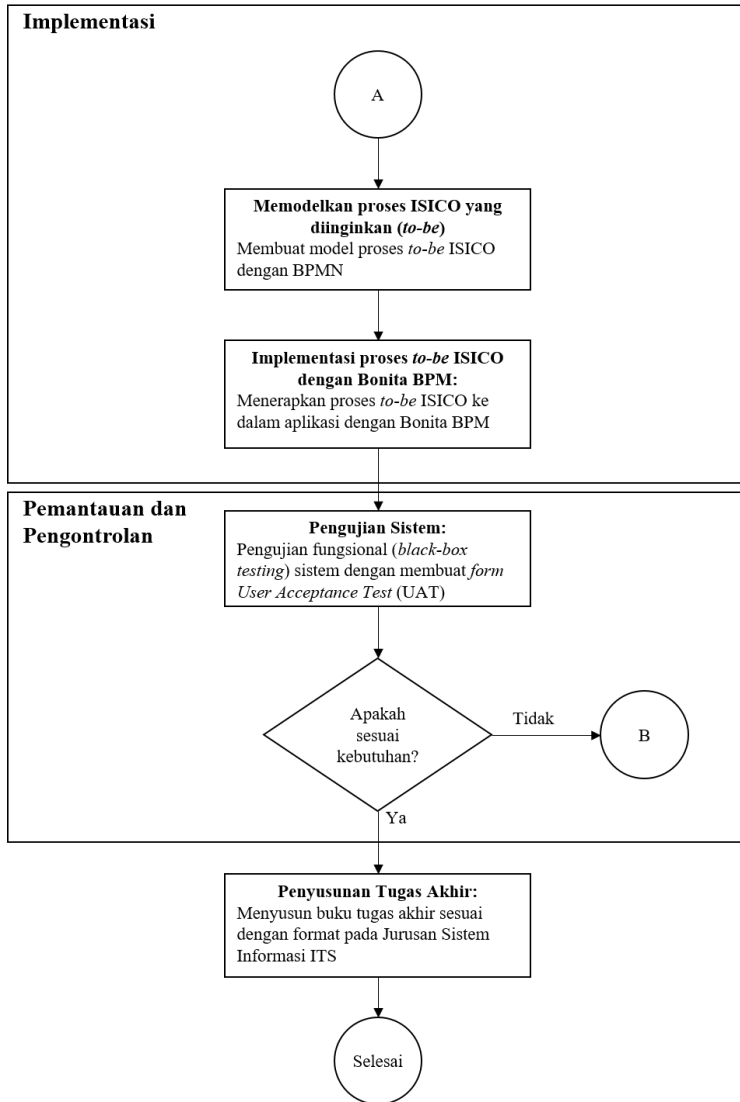
## BAB III

### METODOLOGI

Pada bab ini menjelaskan terkait metodologi yang akan digunakan sebagai panduan untuk menyelesaikan penelitian tugas akhir ini.



Gambar 3.1 Diagram metodologi bagian 1



**Gambar 3.2 Diagram metodologi bagian 2**

### 3.1 Tahapan Pelaksanaan Tugas Akhir

Bagian ini menjelaskan tahapan pelaksanaan tugas akhir yang digambarkan pada Gambar 3.1 dan Gambar 3.2.

#### 3.1.1. Analisis permasalahan

Pada tahap ini, penulis melakukan analisis permasalahan dengan mengetahui kondisi terkini dari proses pendaftaran acara ISICO. Kondisi terkini proses pendaftaran dapat diketahui melalui wawancara dengan penyelenggara ISICO. Untuk mengatasi permasalahan tersebut, penulis mengusulkan melakukan pengembangan sistem berbasis proses. Hasil dari tahap ini adalah gambaran model *as-is* dan *to-be* dengan menggunakan BPMN.

#### 3.1.2. Studi literatur

Setelah mengetahui permasalahan dan mengusulkan solusi, penulis melakukan studi literatur yang didapatkan dari pengumpulan referensi yang bersumber pada buku dan penelitian sebelumnya. Tahap ini dilakukan agar penulis dapat lebih memahami dasar-dasar teori yang berhubungan dengan permasalahan.

#### 3.1.3. Implementasi

Pada tahap ini terbagi menjadi beberapa aktivitas, yaitu identifikasi proses pendaftaran ISICO saat ini (*as-is*), memodelkan proses *as-is*, memodelkan proses *redesign*, dan implementasi ke *software* Bonita BPM.

##### 1. Identifikasi proses saat ini (*as-is*)

Tahap ini diturunkan dari tahap analisis permasalahan, yaitu hasil wawancara yang telah dilakukan selanjutnya diidentifikasi sehingga penulis mendapat gambaran yang lebih jelas lagi terhadap permasalahan proses pendaftaran yang ada.

Hasil yang didapatkan dari tahap ini adalah gambaran jelas tentang proses pendaftaran yang saat ini terjadi di ISICO.

## 2. Memodelkan proses (*as-is*)

Pada tahap ini, penulis mulai memodelkan proses *as-is*, yang didapatkan dari tahap identifikasi proses *as-is*, dengan BPMN pada *software* Bonita BPM. Hal ini dilakukan untuk mempermudah proses berikutnya, yaitu dengan mengganti atau menambahkan beberapa proses sesuai kebutuhan penyelenggara ISICO.

## 3. Memodelkan proses *redesign*

Setelah itu, penulis melakukan pemodelan pada proses *redesign*, yaitu proses-proses yang harus dilakukan oleh pendaftar ISICO. Pemodelan ini diturunkan dari tahap sebelumnya, dari tahap *as-is* dilakukan identifikasi proses mana yang seharusnya tidak perlu atau adanya proses tambahan yang perlu dimasukkan untuk mendukung proses pendaftaran. Pemodelan di sini juga dilakukan dengan BPMN.

## 4. Implementasi ke *software* Bonita BPM

Setelah itu, dilakukan otomasi proses pada model *to-be* tersebut. Hasil akhir dari tahap ini adalah sebuah sistem pendaftaran ISICO yang dapat diakses oleh *user* melalui *website*.

### 3.1.4. Pemantauan dan pengontrolan

Setelah tahap implementasi selesai, dilakukan tahap pemantauan dan pengontrolan. Pada tahap ini, sistem akan dilakukan uji coba fungsional. Target pengujian adalah pengguna sistem, yaitu penyelenggara ISICO. Bila dari hasil pengujian didapatkan kesalahan fungsional atau dari prosesnya, penulis akan mengulangi tahap identifikasi proses

bisnis dan melakukan perbaikan pada sistem. Namun bila dari hasil pengujian sudah tidak terdapat kesalahan, penulis akan melanjutkan ke tahap selanjutnya, yaitu penyusunan buku tugas akhir.

#### 3.1.5. Penyusunan tugas akhir

Tahap ini merupakan tahapan terakhir, yaitu penyusunan dokumen tugas akhir. Format pengerjaan dokumen akan disesuaikan dengan aturan-aturan yang sudah ditetapkan oleh Jurusan Sistem Informasi ITS Surabaya.

*Halaman ini sengaja dikosongkan*

## **BAB IV**

### **PERANCANGAN**

Pada bab empat ini akan membahas mengenai perancangan implementasi pada Bonita BPM, dimulai dari identifikasi proses *as-is* ISICO, pemodelan proses *as-is* ISICO, identifikasi proses *to-be* ISICO, pemodelan proses *to-be* ISICO, dan perancangan implementasi proses *to-be* ISICO pada Bonita BPM. Berikut ini akan dilakukan pembahasan dan penjelasan dari setiap tahapannya.

#### **4.1. Identifikasi proses *as-is* ISICO**

Tahapan ini dilakukan melalui wawancara dengan penyelenggara ISICO di Jurusan Sistem Informasi ITS Surabaya. Wawancara dilakukan dengan tanya-jawab kepada Ibu Amna Shifia selaku panita bagian registrasi di ISICO 2017. Hal-hal yang ditanyakan adalah mengenai alur dari proses pendaftaran yang terjadi pada ISICO.

Berdasarkan hasil wawancara pada lampiran, terdapat empat peran yang ada dalam proses pendaftaran ISICO:

1. Peserta, yaitu sebagai pelanggan dalam proses pendaftaran ISICO. Peserta di sini terbagi menjadi dua jenis, yaitu pemakalah dan non pemakalah. Pemakalah adalah peserta yang mendaftarkan dirinya serta makalah yang telah melewati tahap *review* dari penyelenggara sehingga peserta dapat melakukan presentasi makalah dalam konferensi ini. Non pemakalah adalah peserta yang mendaftarkan dirinya hanya sebagai partisipan atau yang menghadiri konferensi untuk mendapatkan wawasan dari makalah yang disajikan, meningkatkan jaringan, atau alasan-alasan lain.
2. Bagian registrasi, yaitu pihak panitia penyelenggara yang melayani proses pendaftaran peserta mulai dari sebelum hingga saat ISICO berlangsung.

3. Bagian keuangan, yaitu pihak panitia penyelenggara yang mengurus administrasi selama proses pendaftaran ISICO berlangsung.
4. Bagian *editor*, yaitu pihak panitia penyelenggara yang bertugas untuk melakukan pembaharuan informasi terkait makalah peserta yang didaftarkan untuk dipresentasikan dalam acara ISICO.

Selain itu, dari hasil wawancara juga diperoleh bahwa proses yang berjalan dalam pendaftaran ISICO adalah diawali dengan mengirimkan *form* pendaftaran beserta bukti pembayaran yang ditujukan kepada penyelenggara ISICO. Bagian registrasi akan melakukan rekapitulasi pendaftaran peserta dengan menggunakan Google Sheet. Dari sini, proses terbagi menjadi dua berdasarkan pesertanya, yaitu non pemakalah dan pemakalah. Proses beserta identifikasi permasalahan untuk non pemakalah dijelaskan pada Tabel 4.1.

**Tabel 4.1 Identifikasi permasalahan pada *non* pemakalah/partisipan**

No.	Proses	Permasalahan
1.	Peserta mengirim <i>form</i> pendaftaran dan bukti pembayaran melalui <i>email</i> kepada penyelenggara	Terkadang ada <i>email</i> peserta yang terlewat untuk dilihat atau dikonfirmasi
2.	Bagian registrasi merekap data peserta dengan menggunakan Google Sheet	Perekapan melalui Google Sheet membuat bagian lain kesulitan dan terkadang membutuhkan waktu yang lama untuk mencari-cari peserta tertentu saat melakukan verifikasi
3.	Bagian keuangan mengecek bukti	<i>Tidak ada</i>



No.	Proses	Permasalahan
	pembayaran berdasarkan ID peserta	
4.	<p>Terdapat 2 kemungkinan:</p> <ol style="list-style-type: none"> <li>1. Jika bukti pembayaran terverifikasi, bagian keuangan mengeluarkan bukti pelunasan pembayaran (<i>receipt</i>) untuk peserta</li> <li>2. Jika bukti pembayaran tidak terverifikasi, bagian keuangan memberikan kabar kepada bagian registrasi bahwa bukti pembayaran belum terverifikasi</li> </ol>	<i>Tidak ada</i>
5.	Bagian registrasi mengirimkan hasil pengecekan bukti pembayaran peserta kepada peserta yang bersangkutan	<i>Tidak ada</i>
6.	Peserta menerima hasil pengecekan bukti pembayaran peserta dan proses berakhir dengan status peserta terdaftar	<i>Tidak ada</i>
<b>Saat konferensi berlangsung</b>		
7.	Pendaftaran ulang dilakukan secara manual dengan menggunakan	1. Absensi kertas ini menyebabkan penyelenggara

No.	Proses	Permasalahan
	absensi kertas hasil rekapitulasi	<p>membutuhkan alokasi waktu yang cukup lama untuk pelaksanaan pendaftaran ulang</p> <p>2. Hasil rekapitulasi terkadang tidak sesuai dengan status pelunasan biaya pendaftaran peserta sehingga memungkinkan terdapat peserta yang sudah melunasi pembayaran namun namanya belum tercantum</p>

Untuk proses yang dilalui oleh pemakalah dijelaskan pada Tabel 4.2.

**Tabel 4.2 Identifikasi permasalahan pada pemakalah/author**

No.	Proses	Permasalahan
1.	Pemakalah mengunggah dokumen makalah/ <i>paper</i> , <i>form</i> pendaftaran, dan bukti pembayaran melalui <i>email</i> kepada pihak penyelenggara	Terkadang ada <i>email</i> yang terlewat untuk dilihat atau dikonfirmasi
2.	Bagian registrasi merekap data pemakalah dengan menggunakan Google Sheet	Perekapan melalui Google Sheet membuat bagian lain kesulitan dan terkadang membutuhkan waktu yang lama untuk mencari-cari peserta

No.	Proses	Permasalahan
		tertentu saat melakukan verifikasi
3.	<p>Di sini, Bagian registrasi melakukan 2 tugas:</p> <ol style="list-style-type: none"> <li>1) Mengirimkan data pemakalah kepada bagian <i>editor</i> untuk dilakukan verifikasi terhadap jumlah halaman pada makalah</li> <li>2) Mengirimkan data pemakalah kepada bagian keuangan untuk dilakukan verifikasi bukti pembayaran</li> </ol>	<i>Tidak ada</i>
4.	Bagian <i>editor</i> mengecek jumlah makalah dan halaman masing-masing makalah setiap pemakalah	Aktivitas ini terkadang terjadi saat proses <i>camera ready</i> belum selesai sehingga belum bisa dihitung jumlah halaman yang berlebih
5.	<p>Saat pengecekan makalah oleh <i>editor</i> berlangsung, terdapat 2 kemungkinan dari hasil verifikasi:</p> <ol style="list-style-type: none"> <li>1) Membuat catatan terkait jumlah makalah yang terunggah oleh setiap pemakalah benar dan masing-masing makalah tidak terdapat</li> </ol>	<i>Update</i> yang dilakukan oleh <i>editor</i> ini dilakukan pada Google Sheet. Hal ini akan berpengaruh saat registrasi ulang berlangsung di hari pelaksanaan konferensi. Bagian registrasi harus mengecek kembali Google Sheet tersebut dan bila ternyata pembayaran yang

No.	Proses	Permasalahan
	<p>kelebihan halaman</p> <p>2) Membuat notifikasi jumlah makalah yang terunggah oleh setiap pemakalah benar, namun terdapat kelebihan halaman pada makalah</p> <p>Kedua hal tersebut dilakukan pada suatu Google Sheet</p>	<p>dilakukan oleh pemakalah kurang, pemakalah harus melakukan pelunasan terlebih dahulu pada bagian keuangan. Hal ini membuat proses registrasi menjadi lama.</p>
6.	<p>Bagian keuangan melakukan verifikasi bukti pembayaran pemakalah</p>	<p><i>Tidak ada</i></p>
7.	<p>Dari verifikasi bukti pembayaran oleh bagian keuangan menghasilkan hasil sebagai berikut.</p> <p>1) Memberikan notifikasi bahwa pembayaran tidak terverifikasi</p> <p>2) Mengeluarkan bukti pelunasan pembayaran untuk (<i>receipt</i>) pemakalah yang telah melunasi</p>	<p><i>Tidak ada</i></p>
8.	<p>Bagian keuangan mengirimkan hasil verifikasi setiap pemakalah kepada bagian registrasi</p>	<p><i>Tidak ada</i></p>

No.	Proses	Permasalahan
9.	Bagian registrasi mengirimkan hasil verifikasi data pemakalah dan khusus untuk peserta yang terverifikasi terdapat pesan untuk melakukan konfirmasi kehadiran serta kebenaran nama kepada peserta melalui <i>email</i>	<i>Tidak ada</i>
10.	Peserta menerima notifikasi hasil verifikasi data pemakalah	<i>Tidak ada</i>
11.	<p>Di sini terdapat 2 kemungkinan tindakan yang dilakukan peserta:</p> <p>1) Bagi peserta yang dinyatakan pembayaran belum dilakukan atau masih kurang: Bila melakukan pembayaran ulang saat itu, maka proses akan kembali pada No. 2, yaitu bagian registrasi merekap data pemakalah. Namun bila tidak, maka proses berakhir dengan status peserta terdaftar namun pembayaran dilakukan saat konferensi</p>	<i>Tidak ada</i>

No.	Proses	Permasalahan
	berlangsung 2) Bagi peserta yang dinyatakan bahwa pembayarannya terverifikasi, dia menerima bukti pelunasan pembayaran dan pesan konfirmasi kehadiran serta kebenaran nama	
12.	<p>Dari sub proses nomor 2 pada poin 11 terdapat tiga kemungkinan lagi:</p> <ol style="list-style-type: none"> <li>1) Peserta melakukan konfirmasi kehadiran dan kebenaran nama tanpa meminta jenis bukti penerimaan pembayaran lain*</li> <li>2) Peserta melakukan konfirmasi kehadiran dan kebenaran nama serta meminta jenis bukti penerimaan pembayaran lain*</li> <li>3) Peserta melakukan permohonan <i>refund</i> dikarenakan makalahnya ditolak oleh pihak penyelenggara</li> </ol>	<p>Pada proses nomor 12 sub nomor 2, akan membuat pihak penyelenggara kesulitan, dimana ketika peserta menginginkan bukti pelunasan pembayarannya digabung dengan bukti penerimaan pembayaran temannya (<i>one to many receipt</i>)</p>
13.	<p>Dari sub proses nomor 1 pada poin 12, proses berakhir di sini dengan</p>	<p><i>Tidak ada</i></p>

No.	Proses	Permasalahan
	status peserta terdaftar	
14.	Dari sub proses nomor 2 pada poin 12, Bagian registrasi merekap konfirmasi kehadiran dan meneruskan permintaan pergantian bukti pelunasan pembayaran kepada bagian keuangan	<i>Tidak ada</i>
	Bagian keuangan mengubah jenis bukti pelunasan pembayaran sesuai permintaan peserta	<i>Tidak ada</i>
	Bagian keuangan kemudian mengirimkan ulang bukti pelunasan pembayaran tersebut ke bagian registrasi	<i>Tidak ada</i>
	Bagian registrasi menerima bukti pelunasan pembayaran yang baru permintaan peserta	<i>Tidak ada</i>
	Bagian registrasi mengirimkan bukti pelunasan pembayaran peserta kepada peserta yang bersangkutan melalui <i>email</i> dan proses berakhir dengan status peserta terdaftar	<i>Tidak ada</i>
15.	Untuk lanjutan proses 12 sub proses 3, peserta	<i>Tidak ada</i>

No.	Proses	Permasalahan
	melakukan pengisian sebuah <i>form</i> permohonan pengembalian uang ( <i>refund</i> ) yang dikirimnya melalui <i>email</i> .	
	Bagian registrasi akan merekam <i>form</i> tersebut dan mengirimnya pada Bagian Keuangan.	<i>Tidak ada</i>
	Bagian keuangan akan melakukan verifikasi terhadap permohonan tersebut. Bila terverifikasi, maka bagian keuangan akan melakukan <i>refund</i> kepada peserta.	<i>Tidak ada</i>
	Bagian keuangan melakukan konfirmasi <i>refund</i> kepada bagian registrasi untuk kemudian akan diteruskan kepada peserta. Proses berakhir dengan status peserta tidak terdaftar	<i>Tidak ada</i>
	<b>Saat konferensi berlangsung</b>	
16.	Pendaftaran ulang dilakukan secara manual dengan menggunakan absensi kertas	1) Absensi kertas ini menyebabkan penyelenggara membutuhkan alokasi waktu yang cukup lama untuk pelaksanaan pendaftaran ulang



No.	Proses	Permasalahan
		<p>2) Hasil rekapitulasi terkadang tidak sesuai dengan status pelunasan biaya pendaftaran peserta sehingga terdapat peserta yang sudah melunasi pembayaran namun namanya belum tercantum</p> <p>3) Bagian registrasi kesulitan dalam mengecek pelunasan biaya registrasi peserta (pemakalah) karena berkas rekapitulasi yang masih terpisah-pisah</p>
	<b>Saat akan menyampaikan makalah</b>	
17.	Pendaftaran ulang sebelum menyampaikan pemakalah dilakukan secara manual dengan menggunakan absensi kertas	1) Absensi kertas ini menyebabkan penyelenggara membutuhkan alokasi waktu yang cukup lama untuk pelaksanaan pendaftaran ulang

Keterangan:

\*jenis bukti penerimaan pembayaran yang dimaksud pada proses pendaftaran ISICO adalah:

1. *One to one receipt*, dimana satu *receipt* untuk satu nama pemakalah

2. *One to many receipt*, dimana satu *receipt* untuk lebih dari satu nama pemakalah

Dari tabel-tabel di atas, didapat kesimpulan bahwa permasalahan yang terjadi pada proses pendaftaran ISICO saat ini adalah seperti yang ditunjukkan pada Tabel 4.3.

**Tabel 4.3 Kesimpulan permasalahan pada proses pendaftaran ISICO**

No.	Permasalahan
1.	Terkadang ada <i>email</i> peserta yang terlewat untuk dilihat atau dikonfirmasi
2.	Perekapan melalui Google Sheet membuat bagian lain kesulitan dan terkadang membutuhkan waktu yang lama untuk mencari-cari peserta tertentu saat melakukan verifikasi
3.	Absensi kertas saat pelaksanaan konferensi berlangsung menyebabkan penyelenggara membutuhkan alokasi waktu yang cukup lama untuk pelaksanaan pendaftaran ulang
4.	Hasil rekapitulasi terkadang tidak sesuai dengan status pelunasan biaya pendaftaran peserta sehingga terdapat peserta yang sudah melunasi pembayaran namun namanya belum tercantum
5.	Bagian registrasi kesulitan dalam mengecek pelunasan biaya registrasi peserta (pemakalah) karena berkas rekapitulasi yang masih terpisah-pisah
6.	Pihak penyelenggara kesulitan dalam menangani banyak peserta yang melakukan permintaan jenis bukti pelunasan pembayaran lain

#### 4.2. Pemodelan proses *as-is* ISICO

Pada bagian ini akan ditunjukkan pemodelan proses *as-is* ISICO dengan menggunakan *business process modeling notation* (BPMN). Pemodelan dilakukan pada proses sebelum

konferensi berlangsung. Pada Lampiran B dapat dilihat bahwa proses *as-is* ISICO digambarkan dengan dua diagram untuk membedakan proses yang terjadi pada dua jenis peserta, yaitu non pemakalah dan pemakalah. Model *as-is* dengan BPMN ditunjukkan pada Lampiran B Gambar B.1 sampai dengan Gambar B.3 untuk proses pada non pemakalah. Sedangkan pada Lampiran B Gambar B.4 sampai dengan Gambar B.7 menunjukkan model proses pada pemakalah.

Terlihat pada Lampiran B Gambar B.1 sampai dengan Gambar B.3 menunjukkan terdapat 3 *lane* yang menggambarkan aktor pada proses tersebut. Ketiga *lane* tersebut adalah peserta dimana dalam hal ini adalah non pemakalah (partisipan), bagian registrasi, dan bagian keuangan. Lampiran B Gambar B.1 menunjukkan proses yang dimulai dengan peserta mengirimkan *form* pendaftaran dan bukti pembayaran melalui *email* yang digambarkan dengan *human task*. Lalu proses dilanjutkan dengan perekapan pendaftaran oleh bagian registrasi serta meneruskan berkas peserta ke bagian keuangan.

Pada Lampiran B Gambar B.2, bagian keuangan akan melakukan pengecekan bukti pembayaran masing-masing peserta kemudian hasil dari aktivitas tersebut akan menuju ke sebuah *gateway XOR* dimana proses akan berlanjut pada salah satu aktivitas. Jika bukti pembayaran telah terverifikasi, bagian keuangan akan mengeluarkan bukti penerimaan pembayaran (*receipt*) untuk masing-masing peserta. Sebaliknya, bagian keuangan akan memberikan catatan bahwa bukti pembayaran tidak terverifikasi. Pada Lampiran B Gambar B.3, bagian registrasi akan mengirimkan hasil pengecekan bukti pembayaran, beserta *receipt* kepada peserta yang selanjutnya proses berakhir dengan status peserta terdaftar.

Untuk pemakalah (*author*), pada Lampiran B Gambar B.4 menunjukkan bahwa proses dimulai dengan peserta (dalam hal

ini pemakalah) mengirimkan *form* pendaftaran, dokumen makalah, dan bukti pembayaran melalui *email*. Aktivitas ini digambarkan dengan *human task* yang berarti aktivitas yang dilakukan oleh manusia. Kemudian bagian registrasi melakukan rekap pendaftaran yang selanjutnya melalui *gateway* AND pembuka yang berarti dua aktivitas selanjutnya harus dilakukan/diselesaikan bersama sebelum dapat melanjutkan aktivitas pada *gateway* AND penutup. Di sini, bagian registrasi meneruskan/mengirimkan *form* pendaftaran dan bukti pembayaran peserta ke bagian keuangan dan mengirimkan *form* pendaftaran peserta dan dokumen makalah ke bagian *editor*.

Kemudian bagian *editor* akan melakukan pengecekan terhadap makalah tersebut yang kemudian proses akan melalui *gateway* XOR dimana hanya ada satu aktivitas yang akan dijalankan selanjutnya. Hal ini ditunjukkan pada Lampiran B Gambar B.5. Jika makalah memiliki kelebihan halaman, bagian *editor* akan melakukan pembaruan atau penambahan catatan terkait kelebihan halaman pada Google Sheet. Jika tidak, bagian *editor* melakukan pembaruan atau penambahan catatan bahwa tidak ada halaman yang kelebihan. Setelah itu proses akan bertemu pada *gateway* AND penutup untuk selanjutnya adalah bagian keuangan melakukan verifikasi bukti pembayaran. Saat melakukan verifikasi bukti pembayaran, proses akan masuk pada *gateway* XOR dimana hanya satu aktivitas yang akan selanjutnya dijalankan. Jika bukti pembayaran terverifikasi, bagian keuangan akan membuat bukti penerimaan pembayaran (*receipt*) dan jika tidak akan dibuatkan catatan bahwa pembayaran belum terverifikasi.

Pada Lampiran B Gambar B.6, bagi pembayaran yang tidak terverifikasi, bagian registrasi akan memberikan informasi tersebut kepada peserta dan kemudian akan masuk pada *gateway* XOR sebagai pilihan bahwa peserta akan melakukan pembayaran langsung atau tidak. Jika peserta melakukan

pembayaran langsung, proses akan kembali pada perekapan berkas peserta oleh bagian registrasi. Sebaliknya, proses akan berakhir dengan status peserta terdaftar namun pembayaran dilakukan saat konferensi berlangsung. Bagi pembayaran yang telah terverifikasi, bagian registrasi akan mengirimkan *receipt* dan pesan konfirmasi kehadiran peserta dan kebenaran nama. Selanjutnya peserta akan melakukan konfirmasi kehadiran, kebenaran nama, atau meminta jenis *receipt* lain, misalnya meminta penggabungan *receipt* dengan peserta lain. Hal ini bermuara pada *gateway XOR* yang berarti hanya ada satu aktivitas selanjutnya yang dijalankan. Di sini terdapat tiga pilihan dengan 2 kemungkinan, jika peserta tidak memilih jenis *receipt* lain, peserta akan mengisi *form reimburse* yang berarti peserta ingin melakukan pengembalian uang yang dimungkinkan karena makalah/*paper* yang ternyata ditolak, atau peserta melakukan konfirmasi kehadiran dan kebenaran nama. Jika peserta memilih jenis *receipt* lain, peserta melanjutkan aktivitas konfirmasi kehadiran, kebenaran nama, dan meminta jenis *receipt* lain.

Selanjutnya pada Lampiran B Gambar B.7, pada aktivitas pengisian *form reimburse*, bagian registrasi akan melakukan perekapan *form reimburse* yang masuk dan meneruskannya ke bagian keuangan. Lalu bagian keuangan akan melakukan verifikasi terhadap *form reimburse* yang diajukan peserta kemudian mengirimkan bukti pengembalian uang kepada bagian registrasi. Selanjutnya bagian registrasi akan mengirimkannya kepada peserta dan proses akan berakhir dengan status peserta tidak terdaftar yang digambarkan dengan *terminate end event*, yang berarti hal ini merupakan akhir proses yang tidak seharusnya terjadi. Pada peserta yang melanjutkan aktivitas konfirmasi kehadiran dan kebenaran nama, proses akan berakhir dengan status peserta terdaftar. Selanjutnya pada peserta yang memilih ingin mengganti jenis *receipt*, aktivitas dilanjutkan dengan perekapan permintaan

penggantian *receipt* oleh bagian registrasi. Selanjutnya bagian keuangan akan melakukan pengubahan *receipt* dan memberikannya kepada bagian registrasi. Bagian registrasi akan mengirimkan *receipt* yang baru melalui *email* dan proses berakhir dengan status peserta terdaftar.

### 4.3. Identifikasi proses *to-be* ISICO

Tahapan ini juga dilakukan melalui wawancara dengan penyelenggara ISICO di Jurusan Sistem Informasi ITS Surabaya. Wawancara dilakukan dengan tanya-jawab kepada Ibu Amna Shifia selaku panita bagian registrasi di ISICO 2017 dan Bapak Faizal Mahananto selaku Ketua Penyelenggara ISICO 2017. Hal-hal yang ditanyakan adalah mengenai sistem pendaftaran yang diharapkan untuk ISICO.

Dari hasil wawancara didapatkan gambaran tentang sistem pendaftaran yang akan dikembangkan oleh penulis. Di sini, penulis dapat memetakan kebutuhan ISICO dengan permasalahan-permasalahan yang teridentifikasi pada sub bab sebelumnya. Hasil pemetaan ini dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Pemetaan permasalahan dan solusi proses pendaftaran ISICO**

No.	Permasalahan	Solusi
1.	Terkadang ada <i>email</i> peserta yang terlewat untuk dilihat atau dikonfirmasi	Adanya <i>task list</i> dari sistem Bonita BPM
2.	Perekapan melalui Google Sheet membuat bagian lain kesulitan dan terkadang membutuhkan waktu yang lama untuk mencari-cari peserta tertentu saat melakukan verifikasi	<ol style="list-style-type: none"> <li>1. Penyimpanan data peserta menjadi satu dalam sebuah <i>database</i></li> <li>2. Verifikasi dilakukan langsung dalam sistem pada Bonita BPM dan akan</li> </ol>

No.	Permasalahan	Solusi
		memperbarui data dalam <i>database</i>
3.	Absensi kertas menyebabkan penyelenggara membutuhkan alokasi waktu yang cukup lama untuk pelaksanaan pendaftaran ulang	Adanya <i>qr code</i> untuk memudahkan registrasi ulang saat konferensi berlangsung
4.	Hasil rekapitulasi terkadang tidak sesuai dengan status pelunasan biaya pendaftaran peserta sehingga terdapat peserta yang sudah melunasi pembayaran namun namanya belum tercantum	Adanya <i>qr code</i> bagi peserta yang pembayarannya sudah dikonfirmasi
5.	Bagian registrasi kesulitan dalam mengecek pelunasan biaya registrasi peserta (pemakalah) karena berkas rekapitulasi yang masih terpisah-pisah	Adanya <i>qr code</i> yang memberikan informasi tentang status pelunasan biaya pendaftaran peserta
6.	Pihak penyelenggara kesulitan dalam menangani banyak peserta yang melakukan permintaan jenis bukti penerimaan pembayaran yang lain	Peran pendaftar pada sistem Bonita BPM membuat pendaftaran peserta jadi fleksibel. Pendaftar bisa memilih mendaftarkan diri sendiri atau kelompok

Selain itu, pada proses *to-be* di sini, terjadi perubahan peran yang terlibat dalam sistem nantinya. Perubahan tersebut antara lain sebagai berikut:

1. Terdapat peran pendaftar (*registrant*) yang berfungsi dalam mendaftarkan *paper* atau partisipan sehingga untuk pemakalah/*author* maupun peserta non pemakalah digantikan oleh pendaftar ini. Peran ini yang nantinya akan menjalankan sistem. Pendaftar dapat mendaftarkan dirinya sendiri maupun orang lain. Pendaftar juga dapat mendaftarkan *paper* ataupun tidak. Hal ini bertujuan untuk menghindari kesulitan dalam menangani permintaan jenis tagihan dari peserta.
2. Peran registrasi dan *editor* pada proses *as-is* dihilangkan sehingga dalam sistem ini nanti dapat dilakukan oleh pendaftar dan bagian keuangan saja.

Dari hasil pemetaan permasalahan dan solusi pada Tabel 4.4, serta hasil wawancara dan diskusi yang dilakukan (dapat dilihat pada Lampiran C dan Lampiran D) didapatkan gambaran proses *to-be* ISICO seperti pada Tabel 4.5.

**Tabel 4.5 Aktivitas pada proses *to-be* ISICO**

No.	Proses
1.	Peserta melakukan pendaftaran akun pada sistem dengan menggunakan <i>email</i> yang terdaftar pada <i>easy chair</i> (untuk pemakalah) atau untuk yang tidak memiliki akun <i>easychair</i> , dapat mendaftar dengan <i>email</i> apapun.
2.	Peserta memilih apakah ingin mendaftarkan makalah ( <i>paper</i> ) atau tidak.
3.	Jika peserta memilih untuk mendaftarkan <i>paper</i> , peserta mengisi <i>form</i> pendaftaran <i>paper</i> . Dalam <i>form</i> ini, peserta dapat mendaftarkan lebih dari satu <i>paper</i> dan memungkinkan untuk mendaftarkan <i>presenter</i> yang berbeda. Batas maksimal <i>paper</i> yang dapat didaftarkan adalah sebanyak 3 (tiga) <i>paper</i> .
4.	Jika sudah dan/atau peserta tadi tidak memilih



No.	Proses
	mendaftarkan <i>paper</i> , peserta akan menuju ke <i>form</i> dimana peserta dapat memilih, ingin mendaftar partisipan lain atau tidak.
5.	Jika peserta memilih mendaftar partisipan lain, peserta harus mengisi <i>form</i> pendaftaran teman. Teman yang dapat didaftarkan adalah sebanyak 5 (lima) orang.
6.	Jika sudah dan/atau peserta tadi tidak memilih untuk mendaftar partisipan lain, peserta akan menuju ke <i>form</i> dimana peserta dapat memilih untuk membeli produk lain atau tidak. Produk lain di sini berupa prosiding dan layanan <i>booking</i> hotel.
7.	Jika peserta memilih untuk menambah produk lain, peserta akan masuk ke dalam <i>form</i> untuk pendataan berapa prosiding yang ingin dibeli atau mereservasi penginapan hotel.
8.	Jika sudah dan/atau peserta tadi tidak memilih untuk menambah produk lain, peserta telah berhasil melakukan pendaftaran ISICO.
9.	Bagian Keuangan mengirimkan <i>invoice</i> kepada peserta.
10.	Peserta mengunggah bukti pembayaran pada sistem.
11.	Bagian Keuangan meverifikasi bukti pembayaran peserta.
12.	Bagian Keuangan mengirim <i>qrcode</i> dan bukti penerimaan pembayaran kepada peserta.
13.	Peserta yang telah menerima <i>qrcode</i> dan bukti pelunasan dapat melakukan konfirmasi kehadiran.
14.	Jika peserta menyatakan hadir, peserta akan menuju pada halaman untuk konfirmasi kebenaran nama.

No.	Proses
15.	Jika nama yang tertera sudah benar, proses berakhir dengan status peserta terdaftar.
16.	Jika peserta menyatakan tidak hadir, proses berakhir dengan status peserta tidak hadir.
17.	Pada proses nomor 16, bila terdapat kesalahan dalam penulisan nama peserta, peserta akan menuju <i>form</i> pengisian untuk pembenaran nama. Setelah itu, proses berakhir di sini dengan status peserta terdaftar.

#### 4.4. Pemodelan proses *to-be* ISICO

Pada bagian ini akan ditunjukkan pemodelan proses *to-be* ISICO dengan menggunakan *business process modeling notation* (BPMN). Pemodelan proses *to-be* dapat dilihat pada Lampiran E Gambar E.1 sampai dengan Gambar E.3.

Pada Lampiran E Gambar E.1, proses dimulai dengan pendaftar memilih mendaftarkan makalah (*paper*) atau tidak. Kemudian akan masuk pada *gateway* XOR dimana hanya akan ada satu aktivitas yang berjalan selanjutnya. Jika pendaftar memilih mendaftarkan *paper*, pendaftar melakukan penambahan *paper* dan kemudian memilih *presenter* yang akan mempresentasikan *paper* tersebut. Kemudian pada Lampiran E Gambar E.2 terdapat *gateway* XOR lagi untuk pilihan penambahan *paper* tambahan atau tidak. Jika pendaftar memilih mendaftarkan *paper* tambahan, pendaftar melakukan penambahan *paper* tambahan. Jika tidak, pendaftar akan kembali pada aktivitas memilih mendaftarkan *paper* atau tidak. Pada aktivitas menambahkan *paper* tambahan terdapat fungsi *loop* yang berfungsi untuk mengulang aktivitas jika kondisi ingin melakukan penambahan *paper* tambahan terpenuhi. Jika tidak terpenuhi, aktivitas akan berlanjut pada memilih mendaftarkan *paper* atau tidak.

Jika pendaftar memilih tidak mendaftarkan *paper*, pendaftar akan melakukan aktivitas memilih menambah non pemakalah (partisipan) atau tidak yang ditunjukkan pada Lampiran E Gambar E.1. Kemudian akan masuk pada *gateway* XOR. Di sini hanya ada satu aktivitas selanjutnya yang berjalan, yaitu menambahkan partisipan atau lanjut pada aktivitas memilih menambah kebutuhan lain atau tidak. Jika pendaftar menambahkan partisipan, pendaftar akan mengisi *form* pendaftaran partisipan. Pada aktivitas ini terdapat fungsi *loop* yang akan mengulang aktivitas bila penambahan partisipan ingin dilakukan lagi. Jika tidak, pendaftar akan lanjut pada aktivitas selanjutnya, yaitu memilih menambah kebutuhan lain atau tidak seperti pada Lampiran E Gambar E.2.

Di sini kebutuhan lain yang dimaksud adalah pemesan prosiding serta layanan reservasi hotel. Pada aktivitas ini, pendaftar akan masuk pada *gateway* XOR lagi, dengan pilihan pendaftar melakukan pemesan prosiding atau pendaftar lanjut pada aktivitas mengirim pesan registrasi selesai. Jika pendaftar melakukan pemesan prosiding, pendaftar akan masuk pada *gateway* XOR dimana pendaftar memilih akan melakukan reservasi hotel atau langsung melanjutkan pada pengiriman pesan registrasi selesai yang dilakukan dengan pengiriman kepada bagian keuangan dan pendaftar sendiri. Selanjutnya proses akan berlanjut pada bagian keuangan, yaitu mengirimkan *invoice*. Hal ini tergambar pada Lampiran E Gambar E.1. Di sini pengiriman *invoice* akan secara otomatis terkirim pada *email* pendaftar. Selanjutnya pada Lampiran E Gambar E.2, pendaftar melakukan penunggahan bukti pembayaran. Di sini bukti pembayaran akan terkirim pada *email* bagian keuangan. Kembali pada Lampiran E Gambar E.1, bagian keuangan akan melakukan verifikasi bukti pembayaran. Setelah verifikasi selesai, sistem akan secara otomatis melakukan pemberitahuan bahwa pembayaran telah terkonfirmasi kepada pendaftar melalui *email*. Selanjutnya

pada Lampiran E Gambar E.2 bagian keuangan akan mengirimkan tanda bukti penerimaan pembayaran (*receipt*) dan *qr code*. Pengiriman ini juga dilakukan secara otomatis kepada *email* pendaftar. Kemudian pendaftar melakukan konfirmasi kehadiran yang kemudian akan masuk pada *gateway XOR*. Di sini, jika pendaftar melakukan konfirmasi hadir, pendaftar akan melakukan konfirmasi kebenaran nama. Jika tidak, sistem akan secara otomatis mengirimkan *email* kepada pendaftar terkait konfirmasi ketidakhadiran tersebut. Aktivitas ini ditunjukkan dengan *service task* yang berarti sistem yang melakukan aktivitas tersebut. Selanjutnya proses berakhir dengan *terminate end event* yang berstatus peserta tidak hadir.

Pada Lampiran E Gambar E.3, jika pendaftar melakukan konfirmasi kebenaran nama, proses juga akan masuk pada *gateway XOR*. Di sini terdapat pilihan jika semua nama sudah benar, proses akan berakhir dengan status peserta terdaftar. Namun jika tidak, proses akan berlanjut pada aktivitas mengisi *form* pembenaran nama. Pada aktivitas ini terdapat fungsi *loop* yang berfungsi untuk mengulang aktivitas tersebut dengan kondisi jika ingin melakukan pembenaran nama lagi. Jika tidak ingin melakukan pembenaran nama lagi, proses akan berakhir dengan status peserta terdaftar.

#### 4.5. Sistem Eksternal

Pembuatan sistem eksternal pada penelitian ini dilakukan untuk memenuhi kebutuhan penyelenggara ISICO yang tidak dilakukan pada Bonita BPM. Sistem eksternal ini dibangun dengan menggunakan *framework codeigniter*. Beberapa kebutuhan yang dirancang pada sistem eksternal adalah pendaftaran akun Bonita, pembuatan *invoice*, pembuatan bukti penerimaan pembayaran, pembuatan *qr code*, serta informasi seperti jumlah *author* yang terdaftar, jumlah *participant (non author)* yang terdaftar, jumlah peserta yang telah melunasi

pembayaran, dan jumlah peserta yang telah hadir dalam konferensi. Berdasarkan kebutuhan tersebut, dapat dipetakan pengguna sistem eksternal sebagai berikut.

1. Peserta/*registrant*: merupakan pengguna yang memanfaatkan fitur pendaftaran akun Bonita dalam sistem eksternal yang akan dibangun.
2. Bagian keuangan: merupakan pengguna yang melakukan pembuatan *invoice* dan pembuatan *qrcode*.
3. Bagian registrasi: merupakan pengguna yang dapat melakukan konfirmasi kehadiran peserta setelah melakukan *scan qrcode* saat konferensi berlangsung.

Selain pengguna yang telah disebutkan di atas, sistem eksternal ini juga memungkinkan pihak penyelenggara lain untuk mengakses sistem, seperti ketua penyelenggara yang membutuhkan informasi terkait pendaftaran peserta.

#### **4.6. Perancangan Formulir untuk *User Acceptance Testing***

*User acceptance testing* adalah tahap yang dilakukan untuk menguji coba sistem yang telah dibangun untuk mengetahui kesesuaian antara proses yang dimodelkan dengan yang diharapkan. Berikut ini adalah beberapa yang dipersiapkan untuk melakukan uji coba.

##### **4.6.1 *Scope user acceptance testing***

Ruang lingkup dalam melakukan uji coba ini adalah terbatas pada lingkungan internal ISICO, yang mana langsung melibatkan panitia penyelenggara ISICO.

##### **4.6.2 Fitur yang diuji coba**

Fitur yang diuji coba pada tahap ini meliputi semua aktivitas yang ada di dalam sistem. Berikut adalah daftar fitur yang akan diujikan yang dikelompokkan berdasarkan pengguna sistem.

#### 4.6.2.1. Pendaftar

Fitur yang ada pada peserta adalah sebagai berikut.

1. Memilih mendaftarkan *paper* atau tidak
2. Menambahkan *paper*
3. Memilih *presenter*
4. Menambahkan *paper* tambahan (*additional paper*)
5. Memilih menambahkan partisipan atau tidak
6. Menambahkan partisipan
7. Memilih menambah kebutuhan lain atau tidak
8. Memesan prosiding
9. Memesan hotel
10. Mengunggah bukti pembayaran
11. Melakukan konfirmasi kehadiran
12. Melakukan konfirmasi kebenaran nama
13. Mengisi *form* pembenaran nama

#### 4.6.2.2. Bagian Keuangan

Fitur yang ada pada peserta adalah sebagai berikut.

1. Mengirimkan *invoice*
2. Meverifikasi bukti pembayaran
3. Mengirimkan *qr code* dan bukti penerimaan pembayaran

#### 4.6.3 *Responsibilities*

Tabel 4.6 adalah daftar peran yang terlibat dalam pengujian sistem ISICO.

**Tabel 4.6 Peran dalam UAT**

No.	Nama	Peran	<i>Responsibilities</i>
1	Delina Rahayu Effendi	<i>Trainer</i>	Memastikan semua tahap uji coba dilakukan
2	Faizal Mahananto	Peserta	Menjalankan semua fitur yang ada di

			bagian peserta
3	Faizal Mahananto	Bagian Keuangan	Menjalankan semua fitur yang ada di bagian keuangan

#### 4.6.4 Test scripts

Pada tahap ini dilakukan perancangan skenario dalam melakukan uji coba. Skenario ini dibuat sesuai dengan semua fitur yang telah dibangun dalam sistem. Berikut adalah skenario dari uji coba yang dilakukan.

*Script* 1. Tabel 4.7 berisikan skenario mengenai pendaftaran ISICO dengan semua kemungkinan dilakukan.

**Tabel 4.7 Script 1**

<i>Action</i>	<i>Response</i>
<b>Mulai</b>	
Peserta memilih mendaftarkan <i>paper</i>	
Peserta mendaftarkan <i>paper</i> menggunakan ID <i>paper</i>	
	Sistem mengambil data <i>paper</i> sesuai dengan ID <i>paper</i> yang didaftarkan
Peserta memilih <i>presenter</i>	
	Sistem menyimpan pilihan <i>presenter</i> untuk <i>paper</i> terkait
Peserta memilih mendaftarkan <i>paper</i> tambahan	
	Sistem mengambil data

<i>Action</i>	<i>Response</i>
	<i>paper</i> yang dimiliki oleh <i>presenter</i> yang terpilih sebelumnya
Peserta mendaftarkan <i>paper</i> tambahan	
	Sistem menyimpan pendaftaran <i>paper</i> tambahan
Peserta memilih mendaftarkan partisipan	
Peserta mendaftarkan partisipan	
	Sistem menyimpan data partisipan yang didaftarkan
Peserta memilih menambahkan kebutuhan lain	
Peserta memesan prosiding	
	Sistem menyimpan data prosiding yang dipesan
Peserta mereservasi hotel	
	Sistem menyimpan informasi reservasi hotel yang dilakukan
	Sistem mengirimkan pemberitahuan kepada bagian keuangan bahwa ada peserta baru
Bagian keuangan membuat <i>invoice</i> pada sistem	



<i>Action</i>	<i>Response</i>
eksternal	
	Sistem eksternal membuat <i>invoice</i> peserta terkait dalam bentuk Pdf
Bagian keuangan mengirimkan <i>invoice</i>	
	Sistem mengirimkan <i>invoice</i> kepada peserta yang bersangkutan
Peserta mengunggah bukti pembayaran	
	Sistem mengirim pemberitahuan kepada bagian keuangan bahwa peserta terkait telah mengunggah bukti pembayaran
Bagian keuangan meverifikasi bukti pembayaran	
	Sistem menyimpan hasil verifikasi pembayaran
Bagian keuangan membuat <i>qr code</i> untuk masing-masing <i>presenter</i> dan partisipan pada sistem eksternal	
	Sistem eksternal membuat <i>qr code</i> untuk masing-masing <i>presenter</i> dan partisipan yang terdaftar

<i>Action</i>	<i>Response</i>
	dalam <i>invoice</i> terkait
Bagian keuangan membuat bukti penerimaan pembayaran pada sistem eksternal	
	Sistem eksternal membuat bukti penerimaan pembayaran peserta terkait dalam bentuk Pdf
Bagian keuangan mengirim bukti penerimaan pembayaran dan <i>qrcode</i>	
	Sistem mengirimkan pemberitahuan kepada peserta terkait bahwa bukti penerimaan pembayaran dan <i>qrcode</i> dikirimkan
Peserta konfirmasi kehadiran	
Peserta konfirmasi kebenaran nama	
	Sistem mengirimkan pesan kepada peserta bahwa serangkaian aktivitas pendaftaran peserta ISICO telah dilakukan
<b>Selesai</b>	

*Script 2.* Tabel 4.8 berisikan pendaftaran *paper* utama namun melakukan konfirmasi tidak hadir.

Tabel 4.8 Script 2

<i>Action</i>	<i>Response</i>
<b>Mulai</b>	
Peserta memilih mendaftarkan <i>paper</i>	
Peserta mendaftarkan <i>paper</i> menggunakan ID <i>paper</i>	
	Sistem mengambil data <i>paper</i> sesuai dengan ID <i>paper</i> yang didaftarkan
Peserta memilih <i>presenter</i>	
	Sistem menyimpan pilihan <i>presenter</i> untuk <i>paper</i> terkait
Peserta memilih tidak mendaftarkan partisipan	
Peserta memilih tidak menambahkan kebutuhan lain	
	Sistem mengirimkan pemberitahuan kepada bagian keuangan bahwa ada peserta baru
Bagian keuangan membuat <i>invoice</i> pada sistem eksternal	
	Sistem eksternal membuat <i>invoice</i> peserta terkait dalam bentuk Pdf
Bagian keuangan	

<i>Action</i>	<i>Response</i>
mengirimkan <i>invoice</i>	
	Sistem mengirimkan <i>invoice</i> kepada peserta yang bersangkutan
Peserta mengunggah bukti pembayaran	
	Sistem mengirim pemberitahuan kepada bagian keuangan bahwa peserta terkait telah mengunggah bukti pembayaran
Bagian keuangan meverifikasi bukti pembayaran	
	Sistem menyimpan hasil verifikasi pembayaran
Bagian keuangan membuat <i>qrcode</i> untuk masing-masing <i>presenter</i> dan partisipan pada sistem eksternal	
	Sistem eksternal membuat <i>qrcode</i> untuk masing-masing <i>presenter</i> dan partisipan yang terdaftar dalam <i>invoice</i> terkait
Bagian keuangan membuat bukti penerimaan pembayaran pada sistem eksternal	

<i>Action</i>	<i>Response</i>
	Sistem eksternal membuat bukti penerimaan pembayaran peserta terkait dalam bentuk Pdf
Bagian keuangan mengirim bukti penerimaan pembayaran dan <i>qrcode</i>	
	Sistem mengirimkan pemberitahuan kepada peserta terkait bahwa bukti penerimaan pembayaran dan <i>qrcode</i> dikirimkan
Peserta memilih konfirmasi tidak hadir dalam konferensi	
	Sistem mengirimkan pesan kepada peserta terkait konfirmasi ketidakhadirannya
<b>Selesai</b>	

Script 3. Tabel 4.9 berisikan skenario pendaftaran partisipan dengan melakukan pembenaran nama.

**Tabel 4.9 Script 3**

<i>Action</i>	<i>Response</i>
<b>Mulai</b>	
Peserta memilih tidak mendaftarkan <i>paper</i>	
Peserta memilih mendaftarkan partisipan	
Peserta mendaftarkan	

<i>Action</i>	<i>Response</i>
partisipan	
	Sistem menyimpan data partisipan yang didaftarkan
	Sistem mengirimkan pemberitahuan kepada bagian keuangan bahwa ada peserta baru
Bagian keuangan membuat <i>invoice</i> pada sistem eksternal	
	Sistem eksternal membuat <i>invoice</i> peserta terkait dalam bentuk Pdf
Bagian keuangan mengirimkan <i>invoice</i>	
	Sistem mengirimkan <i>invoice</i> kepada peserta yang bersangkutan
Peserta mengunggah bukti pembayaran	
	Sistem mengirim pemberitahuan kepada bagian keuangan bahwa peserta terkait telah mengunggah bukti pembayaran
Bagian keuangan meverifikasi bukti pembayaran	
	Sistem menyimpan hasil

<i>Action</i>	<i>Response</i>
	verifikasi pembayaran
Bagian keuangan membuat <i>qrcode</i> untuk masing-masing <i>presenter</i> dan partisipan pada sistem eksternal	
	Sistem eksternal membuat <i>qrcode</i> untuk masing-masing <i>presenter</i> dan partisipan yang terdaftar dalam <i>invoice</i> terkait
Bagian keuangan membuat bukti penerimaan pembayaran pada sistem eksternal	
	Sistem eksternal membuat bukti penerimaan pembayaran peserta terkait dalam bentuk Pdf
Bagian keuangan mengirim bukti penerimaan pembayaran dan <i>qrcode</i>	
	Sistem mengirimkan pemberitahuan kepada peserta terkait bahwa bukti penerimaan pembayaran dan <i>qrcode</i> dikirimkan
Peserta melakukan konfirmasi kehadiran	
Peserta memilih untuk melakukan pembenaran	

<i>Action</i>	<i>Response</i>
nama	
Peserta mengisi <i>form</i> pembenaran nama	
	Sistem menyimpan data pembenaran nama
	Sistem mengirimkan pesan kepada peserta bahwa serangkaian aktivitas pendaftaran peserta ISICO telah dilakukan
<b>Selesai</b>	

#### 4.6.5 *Checklist item*

*Checklist item* berisikan informasi mengenai berjalan atau tidaknya suatu fungsi *connector* atau aktivitas pada sistem. Tabel 4.10 merupakan *checklist item* keberhasilan aktivitas pada proses, Tabel 4.11 *checklist* keberhasilan *messaging connector* yang berupa notifikasi *email* selama proses berjalan, Tabel 4.12 merupakan *checklist item* keberhasilan fungsional *database*, Tabel 4.13 merupakan *checklist item* keberhasilan fungsional sistem eksternal.

**Tabel 4.10** *Checklist item* keberhasilan aktivitas pada proses

No.	Aktivitas	<i>Pass</i>	<i>Fail</i>
	<b>Peserta</b>		
1	Memilih mendaftarkan <i>paper</i> atau tidak		
2	Menambahkan <i>paper</i>		
3	Memilih <i>presenter</i>		



4	Menambahkan <i>paper</i> tambahan		
5	Memilih menambahkan partisipan atau tidak		
6	Menambahkan partisipan		
7	Memilih menambahkan kebutuhan lain atau tidak		
8	Memesan prosiding		
9	Mereservasi hotel		
10	Mengunggah bukti pembayaran		
11	Melakukan konfirmasi kehadiran		
12	Melakukan konfirmasi kebenaran nama		
13	Mengisi <i>form</i> pembenaran nama		
<b>Bagian Keuangan</b>			
14	Mengirimkan <i>invoice</i>		
15	Meverifikasi bukti pembayaran		
16	Mengirimkan <i>qrcode</i> dan bukti penerimaan pembayaran		

Keterangan:

*Pass* : Aktivitas tidak dapat berjalan

*Fail* : Aktivitas dapat berjalan

**Tabel 4.11 Checklist item keberhasilan notifikasi email**

No.	Pemberitahuan	<i>Pass</i>	<i>Fail</i>
1	Peserta baru kepada peserta		
2	Peserta baru kepada bagian keuangan		

3	<i>Invoice</i> terkirim kepada peserta		
4	Bukti pembayaran terkirim kepada bagian keuangan		
5	Pembayaran terkonfirmasi kepada peserta		
6	<i>Qrcode</i> dan bukti penerimaan pembayaran terkirim kepada peserta		
7	Pesan ketidakhadiran kepada peserta		
8	Proses pendaftaran ISICO selesai kepada peserta		

Keterangan:

*Pass* : Pemberitahuan berjalan dengan normal

*Fail* : Pemberitahuan tidak dapat berjalan

**Tabel 4.12 Checklist item keberhasilan fungsional database**

No.	Fungsi	<i>Pass</i>	<i>Fail</i>
1	Memperbarui <i>paper</i> yang terdaftar berdasarkan ID <i>paper</i>		
2	Mengambil <i>membership presenter</i>		
3	Mengambil <i>presenter</i> berdasarkan ID <i>paper</i> yang terdaftar		
4	Mengambil judul <i>paper</i> berdasarkan ID <i>paper</i> yang terdaftar		
5	Memperbarui <i>presenter paper</i> yang dipilih		

No.	Fungsi	Pass	Fail
6	Mengambil jumlah sisa <i>paper</i> pada suatu <i>paper</i> (saat peserta melakukan pendaftaran <i>paper</i> tambahan)		
7	Mengambil judul <i>paper</i> sesuai dengan <i>presenter</i> yang terpilih (saat peserta melakukan pendaftaran <i>paper</i> tambahan)		
8	Memperbarui <i>presenter paper</i> tambahan yang didaftarkan		
9	Menyimpan data partisipan		
10	Menyimpan data pemesanan prosiding		
11	Mengambil data hotel untuk layanan reservasi hotel		
12	Menyimpan data reservasi hotel		
13	Memperbarui bahwa <i>invoice</i> telah lunas		
14	Memperbarui bahwa <i>paper</i> yang tertulis pada <i>invoice</i> telah lunas		
15	Memperbarui bahwa partisipan yang tertulis pada <i>invoice</i> telah lunas		
16	Menyimpan data pembenaran nama		

Keterangan:

*Pass* : Fungsi berjalan dengan normal

*Fail* : Fungsi tidak dapat berjalan

Tabel 4.13 *Checklist item fungsional sistem eksternal*

No.	Aktivitas	Pass	Fail
1	Pendaftaran akun Bonita		
2	Pembuatan <i>invoice</i>		
3	Pembuatan bukti penerimaan pembayaran		
4	Pembuatan <i>qrcode</i>		
5	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang terdaftar		
6	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang telah hadir		
7	<i>Dashboard</i> informasi jumlah partisipan yang terdaftar		
8	<i>Dashboard</i> informasi jumlah partisipan yang telah hadir		
9	<i>Dashboard</i> informasi jumlah <i>invoice</i> yang telah lunas		

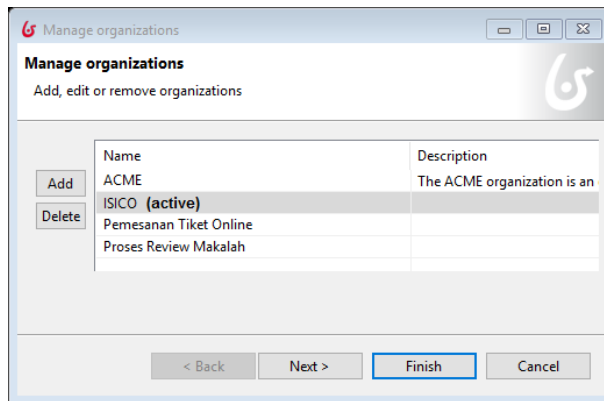
## BAB V

### IMPLEMENTASI

Dalam tahap implementasi ini berisi tentang hasil dari proses perancangan yang telah dilakukan pada bab IV yang diimplementasikan ke dalam aplikasi Bonita BPM. Berikut ini adalah tahapan dalam proses implementasi hingga terbentuk suatu sistem pendaftaran.

#### 5.1. Penentuan Aktor

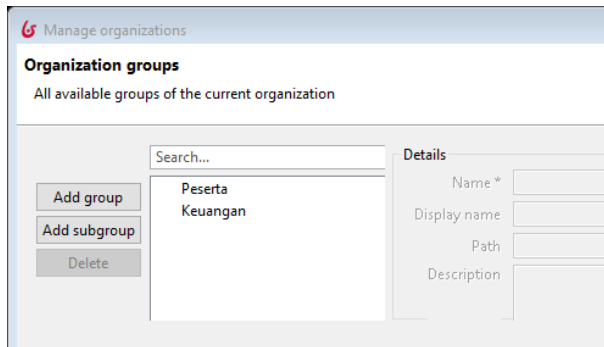
Aktor digambarkan melalui sebuah *lane* yang di dalamnya memiliki aktivitas-aktivitas pada proses yang digambarkan dalam BPMN. Dalam menentukan aktor, perlu ditentukan siapa saja yang terlibat dalam proses. Pada studi kasus ini, terdapat dua aktor yang terlibat, yaitu peserta (baik pemakalah maupun non pemakalah) dan bagian keuangan. Berikut ini merupakan cara membuat dan menentukan aktor dalam Bonita BPM:



**Gambar 5.1** *Manage Organization*

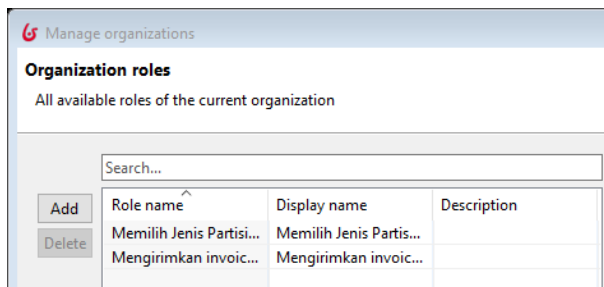
Masuk pada menu *Organization* lalu pilih sub-menu *Manage* hingga muncul kotak dialog *Manage Organizations* seperti pada Gambar 5.1 di atas. Kemudian pilih organisasi yang menjalani proses. Dalam kasus ini adalah ISICO. Jika belum

ada, tambahkan organisasi dengan menekan tombol *Add* dan masukkan nama yang sesuai.



**Gambar 5.2 Manage group**

Kemudian menambahkan grup untuk memetakan pengguna yang ada berdasarkan perannya masing-masing seperti yang terlihat pada Gambar 5.2. Grup di sini sama seperti aktor. Untuk studi kasus ini, aktornya adalah peserta dan keuangan.



**Gambar 5.3 Manage role**

Selanjutnya Gambar 5.3 adalah menambahkan *role*. *Role* di sini berfungsi sama seperti *group*, yaitu untuk memetakan aktivitas-aktivitas yang dapat dijalankan pengguna sesuai dengan perannya. Contohnya adalah memilih mendaftarkan *paper* atau tidak dan mengirimkan *invoice*.

**Gambar 5.4 Setting user**

Selanjutnya adalah membuat *user* seperti pada Gambar 5.4 di atas. Ada beberapa hal yang wajib diisi pada *user* seperti *username*, *password*, dan *membership*. Pada *membership* ini bertujuan untuk memetakan *user* tersebut pada *group* dan *role* yang telah ditentukan pada langkah-langkah sebelumnya. Penentuan ini disesuaikan dengan perannya pada sistem yang akan dibangun. Gambar 5.5 di bawah ini.

	First name	Last name	Username ^
	Delina	Rahayu	delinare
	Ipung		keuangan

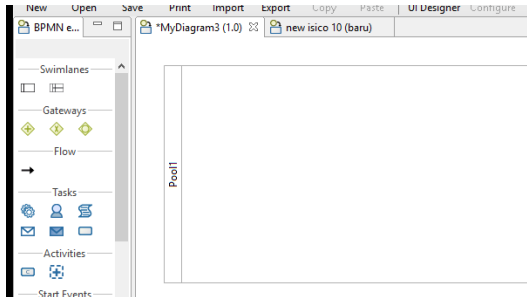
**Gambar 5.5 Manage user**

## 5.2. Pembuatan Proses Model

Pada sub bab pembuatan proses model ini akan menjelaskan bagaimana menggambarkan proses *to be* yang telah dirancang sebelumnya pada aplikasi Bonita BPM. Secara umum dalam melakukan penggambaran ini adalah dengan *drag and drop* atribut yang dibutuhkan ke dalam *workspace*. Berikut adalah langkah-langkah dalam menggambarkan/memasukkan proses ke dalam aplikasi.

### 5.2.1. Menambahkan *pool*

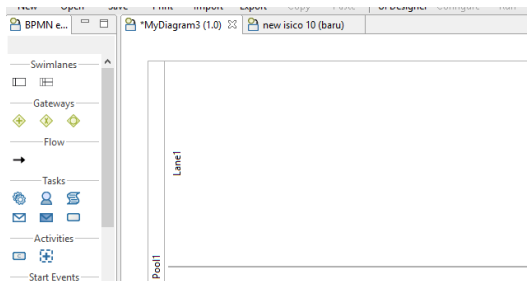
Untuk menambahkan *pool* ke dalam *workspace*, pilih gambar *pool* pada atribut *Swimlanes* yang ada pada sisi bagian kiri layar Bonita. Hasil penambahan *pool* terlihat seperti Gambar 5.6 di bawah ini.



**Gambar 5.6 Menambahkan *pool***

### 5.2.2. Menambahkan *lane*

Penambahan *lane* dilakukan apabila diperlukan untuk mengelompokkan aktivitas yang ada sesuai dengan *group/aktor* yang terlibat. Atribut ini bisa didapatkan pada kelompok atribut *swimlanes*, yaitu tepat di sebelah *pool*. Untuk menambahkannya dapat dilakukan langsung dengan *drag and drop* pada *pool* yang telah ada pada *workspace*. Hasil penambahan *lane* terlihat seperti Gambar 5.7 di bawah ini.

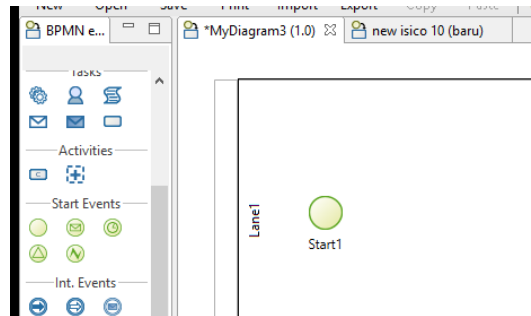


**Gambar 5.7 Menambahkan *lane***



### 5.2.3. Menambahkan *start events*

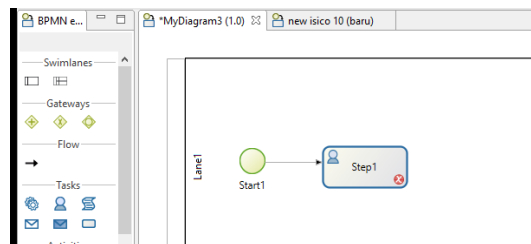
Atribut ini digunakan untuk menandai bahwa sebuah proses sedang berjalan atau tidak. Atribut ini bisa dipilih pada kelompok atribut *start events* lalu ditambahkan ke dalam *workspace*. Penambahan *start events* terlihat pada Gambar 5.8 di bawah ini.



**Gambar 5.8 Menambahkan *start events***

### 5.2.4. Menambahkan *task* dan *flow*

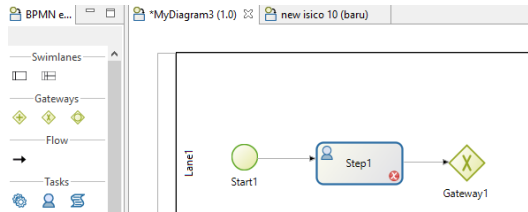
*Task* di sini akan menggambarkan aktivitas-aktivitas yang berjalan pada proses. Untuk menambahkan *task*, pilih atribut aktivitas yang sesuai dengan kebutuhan lalu ditambahkan ke dalam *workspace*. Selain itu, tambahkan juga atribut *transition* pada kelompok atribut *flow* untuk menandakan perpindahan aktivitas yang satu dengan lainnya. Hasil penambahan *task* dan *flow* terlihat seperti Gambar 5.9 di bawah ini.



**Gambar 5.9 Menambahkan *task* dan *flow***

### 5.2.5. Menambahkan gateway

Untuk menambahkan *gateway* dilakukan dengan memilih jenis *gateway* yang dibutuhkan pada kelompok atribut *gateways* lalu dimasukkan ke dalam *workspace*. Penambahan *gateway* terlihat seperti Gambar 5.10 di bawah ini.



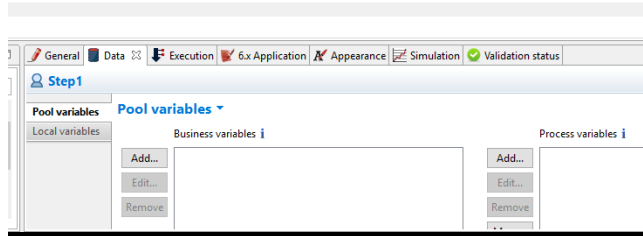
**Gambar 5.10 Menambahkan gateway**

## 5.3. Pembuatan Variabel

Pembuatan variabel dilakukan untuk menyimpan data pada suatu aktivitas dalam Bonita BPM yang dapat digunakan atau dimunculkan pada aktivitas lain bila dibutuhkan. Dalam Bonita BPM ini terdapat tiga jenis variabel, yaitu variabel proses, variabel global, dan variabel lokal. Variabel proses merupakan variabel yang dapat digunakan pada proses keseluruhan atau beberapa aktivitas/*task* saja. Pada penelitian ini digunakan jenis variabel proses saja karena hal ini memungkinkan variabel tertentu dalam suatu aktivitas dapat digunakan pada aktivitas lain.

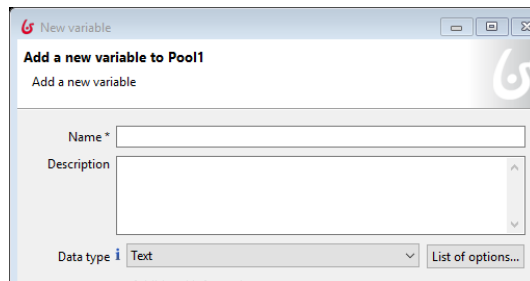
Penggunaan variabel global hampir sama dengan variabel proses, hanya saja variabel global dapat digunakan pada banyak proses dengan konsep *business data model*. Variabel ini dibutuhkan apabila kita memiliki lebih dari satu proses dengan model data yang sama atau kita menggambar *form* sebuah proses dengan menggunakan UI Designer. Berbeda dengan variabel lokal yang penggunaannya lebih sempit, yaitu digunakan dan dibutuhkan pada satu *task*/aktivitas saja.

Untuk membuat variabel hal pertama yang dilakukan adalah masuk pada properti menu *Data* lalu pilih *Pool variables* seperti yang ditunjukkan pada Gambar 5.11 di bawah ini.



**Gambar 5.11 Menu Data**

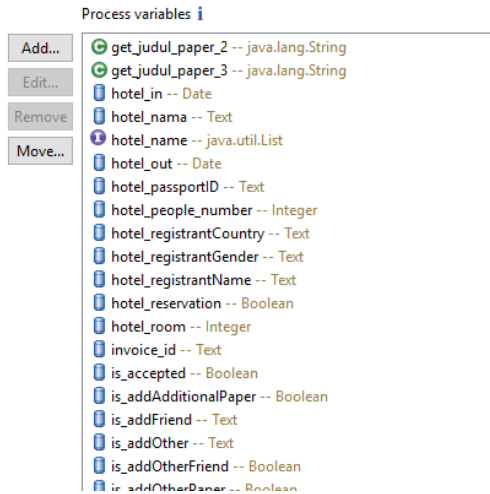
Kemudian pada variabel proses, tekan tombol *Add* untuk menambahkan variabel hingga muncul kotak dialog “*New variable*” seperti pada Gambar 5.12. Dalam kotak dialog tersebut harus diisikan informasi variabel berupa nama serta pilih tipe datanya. Pada tipe data tersebut, disediakan pula tipe data *java object* yang jenisnya dapat dipilih melalui tombol *list of options* di sebelahnya.



**Gambar 5.12 Menambahkan variabel proses**

Gambar 5.13 menunjukkan beberapa variabel yang digunakan dalam penelitian ini. Pada penelitian ini, penulis menggunakan variabel proses biasa serta variabel proses yang memiliki tipe Java Object. Variabel bertipe Java Object ini digunakan saat sistem berinteraksi dengan *database* eksternal, MySQL,

dimana untuk menyimpan data yang telah diambil dari *database* eksternal ke variabel bertipe Java Object ini.



**Gambar 5.13 Variabel yang digunakan**

## 5.4. Pembuatan *Form*

*Form* di sini bertujuan sebagai *user interface* yang menghubungkan pengguna dengan sistem di dalamnya. Dengan *form* ini, sistem dapat menyimpan masukan dari pengguna ke dalam variabel yang telah dideklarasikan.

Dalam penelitian kali ini, penulis menggunakan *form* 6.x yang merupakan pengembangan tampilan yang telah disediakan oleh Bonita BPM. Penggunaan *form* jenis ini dikarenakan penulis menggunakan variabel proses.

### 5.4.1. Formulir Peserta

Formulir peserta di sini meliputi semua jenis formulir yang nantinya akan digunakan untuk peserta, atau disebut registrant. Cara untuk membuat formulir ini adalah dengan masuk ke dalam menu 6.x *application* yang terdapat pada panel di

bawah lalu klik tombol *Add...* pada *tab Pageflow* hingga muncul tampilan *Add form* seperti Gambar 5.14 di bawah ini.

**Gambar 5.14 Tampilan *add form***

Pada Gambar 5.15 dapat juga dipilih variabel mana saja yang akan digunakan untuk *form* itu. Karena di sini penulis menggunakan *process variables* maka untuk memilih variabel yang akan digunakan masuk ke dalam *tab process variables* lalu mencentang semua variabel yang dibutuhkan. Namun, sebenarnya pengguna juga bisa menambahkan variabel ketika *form* sudah terbentuk. Penambahan variabel bila dilakukan di awal seperti ini hanya mempermudah terbentuknya *form* dan konfigurasi masing-masing *widget* pada *form*, sehingga pengguna tidak perlu melakukan konfigurasi lagi. Selain itu, dapat juga memilih apakah *widgetnya* nanti *mandatory* atau tidak serta *read only* atau tidak.

Name	Widget		Mandatory	Read only
<input checked="" type="checkbox"/> is_addPaper	Text field	▼	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> is_addFriend	Text field	▼	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> is_addOther	Text field	▼	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> is_present	Text field	▼	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> is_rightname	Text field	▼	<input type="checkbox"/>	<input type="checkbox"/>

**Gambar 5.15 Menentukan variabel yang digunakan**

Langkah tersebut adalah yang dilakukan ketika akan membuat *form* menggunakan *6.x application*. Berikut ini adalah yang penulis lakukan dalam membuat *form* di masing-masing aktivitas.

### 1. Memilih Mendaftarkan *Paper* Atau Tidak

Proses pendaftaran ISICO ini dimulai dengan peserta atau *registrant* memilih mendaftar *paper* atau tidak. Di sini peserta hanya melakukan pemilihan saja untuk dapat lanjut ke aktivitas selanjutnya.

The image shows a web form interface. At the top, there is a label 'username' followed by a text input field. Below this is a large rectangular area containing the text '<html>'. Underneath this area are two radio buttons, each followed by a horizontal line for a label. At the bottom of the form is a 'Submit' button.

**Gambar 5.16** Formulir memilih mendaftar *paper* atau tidak

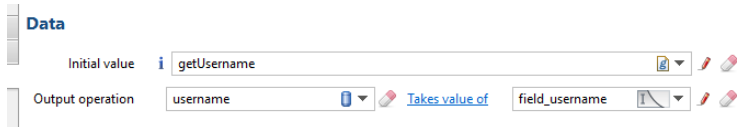
Pada *form* ini, penulis menggunakan *hidden widget* untuk mengambil *username* peserta saat masuk ke dalam sistem. Hal ini berguna saat melakukan penyimpanan data peserta ke dalam *database*. Untuk *hidden widget* ini karena mengambil *username* dengan konfigurasi yang ada di Bonita, maka menggunakan *initial value* bertipe *script*. Hal ini dilakukan dengan masuk ke *tab General* dan lalu *tab Data*. Pada *initial value* klik ikon pensil yang terletak di sebelah kanannya. *Script* untuk mengambil *username* dapat dilihat pada Gambar 5.17 berikut.

```
1. BonitaUsers.getUser(apiAccessor,
   loggedUserId).getUserName()
```

**Gambar 5.17** *Script* mengambil *username* pada Bonita

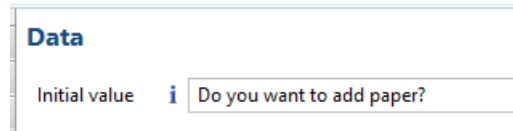
Pada baris pertama tersebut menunjukkan bahwa sistem akan mengambil *username* yang sedang menjalankan proses. Untuk dapat mengambil *username* yang sedang menjalankan

proses, Bonita telah menyediakan variabelnya, yaitu *apiAccessor* dan *loggedUserId*. Variabel tersebut didapatkan dengan membuka *dropdown-menu Select a provided variable*. Untuk *widget* ini, variabel disimpan pada variabel proses *username*, dimana *value*-nya diambil dari *field\_username* seperti yang terlihat pada Gambar 5.18 di bawah ini.



**Gambar 5.18 Konfigurasi penyimpanan pada widget**

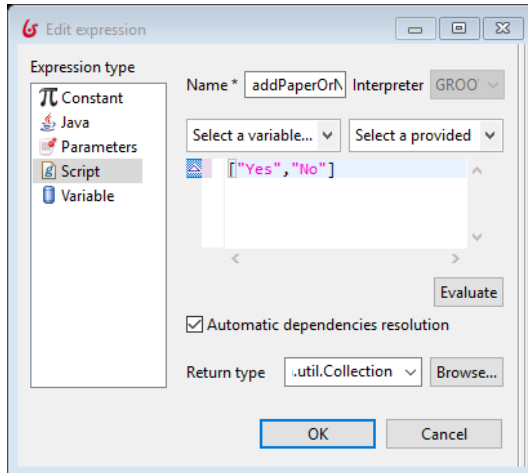
Selanjutnya ada HTML *widget* yang digunakan penulis untuk menampilkan tulisan. *Widget* ini tidak membutuhkan variabel, sehingga konfigurasi pada *tab Data* hanya berisi tulisan yang akan ditampilkan seperti yang terlihat pada Gambar 5.19.



**Gambar 5.19 Konfigurasi HTML aktivitas memilih *paper* atau tidak**

Untuk pemilihan menambahkan *paper* atau tidak digunakan *widget radiobutton*. Konfigurasinya adalah dengan memberikan *available values* pada *tab General* bagian *Data*, lalu masuk ke *edit expression* sama seperti sebelumnya. Kemudian tuliskan pilihan yang mungkin untuk dipilih peserta dalam aktivitas seperti yang terlihat pada Gambar 5.20.

Selanjutnya terdapat *widget Submit* untuk menyimpan data dan berpindah ke aktivitas berikutnya. Gambar 5.21 merupakan tampilan formulir aktivitas memilih menambahkan *paper* atau tidak apabila proses dijalankan.



**Gambar 5.20 Konfigurasi *radiobutton***

### **Paper Registration**

Do you want to add paper?

☐ Yes  
☐ No

**SUBMIT**

**Gambar 5.21 Tampilan formulir memilih menambahkan *paper* atau tidak**

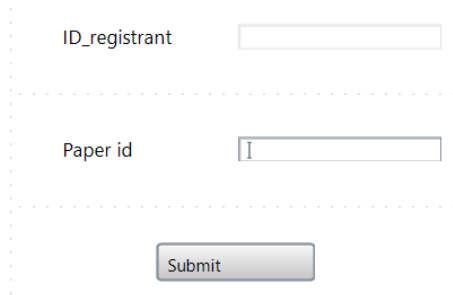
## **2. Add Paper**

Penambahan *paper* ini merupakan aktivitas pilihan yang akan dilakukan peserta ketika memilih akan menambahkan *paper* pada aktivitas sebelumnya. Penambahan *paper* dilakukan dengan memasukkan nomor ID *paper* yang didapatkan dari proses seleksi *paper* melalui *easychair*. Gambar di 5.22 adalah *widget* yang digunakan di dalam *form Add paper*.

Pada Gambar 5.22, terdapat dua *widget*, yaitu *hidden widget* untuk ID *registrant* yang fungsinya kali ini digunakan untuk



menyimpan *paper* ke dalam *database* eksternal Bonita sesuai dengan nomor ID *registrant*.

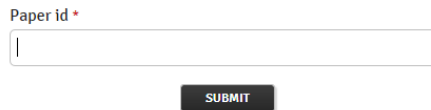


The form consists of two input fields and a submit button. The first field is labeled 'ID\_registrant' and the second is labeled 'Paper id'. Both fields are empty. Below the fields is a button labeled 'Submit'.

**Gambar 5.22 Form menambahkan *paper***

Sedangkan untuk *paper* ID menggunakan *field widget*, yang masukannya nanti akan disimpan pada variabel *paper\_id*.

#### **Add Paper**



The form is titled 'Add Paper'. It features a single input field labeled 'Paper id \*' with a red asterisk indicating it is mandatory. Below the field is a button labeled 'SUBMIT'.

**Gambar 5.23 Tampilan formulir *add paper***

Dalam Gambar 5.23 di atas merupakan tampilan formulir aktivitas menambahkan *paper* ketika dijalankan. Peserta harus mengisikan *paper* ID agar dapat lanjut ke aktivitas berikutnya, karena data tersebut bersifat *mandatory*.

### **3. Memilih *Presenter***

Setelah mendaftarkan *paper* melalui nomor ID nya, peserta harus menetapkan *presenter* untuk mempresentasikan *paper* terkait. Pilihan *presenter* yang ada ini berdasarkan *author* yang berkontribusi pada *paper* terkait. Gambar 5.24 adalah *form* yang digunakan dalam memilih *presenter*.

Pada Gambar 5.24, *widget* yang digunakan untuk *paper* ID dan *paper title* adalah *field widget*. Kedua *widget* ini bersifat

*read only* dengan data yang sudah terisi berdasarkan masukan pada aktivitas *add paper*. Untuk *select presenter* dan status menggunakan *select widget* dengan data yang diambil dari *database* eksternal. Untuk menampilkan data dari *database* eksternal ini harus menggunakan pilihan tipe variabel Java Object dan memilih tipe Java Util List untuk variabel yang memiliki banyak pilihan (*list*). *Widget* terakhir yang digunakan adalah *checkbox widget* untuk memberikan pilihan kepada peserta terkait penambahan *paper* dengan *presenter* yang sama seperti yang dipilih dalam formulir ini.

Gambar 5.25 merupakan tampilan dalam memilih *presenter*. Di sini, peserta harus mengisikan *select presenter* dan *select status*, sedangkan pilihan akan menambahkan *paper* tambahan tidak bersifat *mandatory*.

Paper id

.....

Paper title

.....

Select presenter

.....

Status

.....

Add additional pa... ☒

.....

**Gambar 5.24** Formulir memilih *presenter*

**Set Presenter**

Paper id

Paper title

Select presenter \*

Status \*

Add additional paper  
☐

**SUBMIT**

**Gambar 5.25 Tampilan formulir memilih *presenter***

#### 4. *Add Additional Paper*

Aktivitas ini terjadi jika peserta mencentang pernyataan *checkbox Add additional paper* pada aktivitas memilih *presenter* sebelumnya. Gambar 5.26 merupakan *formulir* untuk menambahkan *paper* tambahan dengan *presenter* yang telah terpilih pada aktivitas sebelumnya.

Paper presenter

.....

Remaining paper

.....

Select Paper

.....

Add other paper ☒

.....

**Submit**

**Gambar 5.26 Formulir menambahkan *paper* tambahan**

Pada 5.26, *paper presenter* dan *remaining paper* merupakan *widget* yang bersifat *read only* saja. *Remaining paper* di sini memberikan informasi tentang sisa *paper* yang dimiliki oleh *presenter* terkait, dimana bila sisanya sudah menunjukkan angka 0 (nol), peserta disarankan untuk melanjutkan aktivitas tanpa harus mencentang *add other paper*. Pilihan tersebut digunakan bila *remaining paper* untuk *presenter* terkait masih lebih dari 0 (nol) yang berarti *presenter* tersebut akan mempresentasikan lebih dari satu *paper*.

Pada tampilan Gambar 5.27 di bawah ini, peserta harus mengisikan atau memilih *paper* pada *widget select paper* karena data ini bersifat *mandatory*. Namun peserta tidak wajib mencentang pernyataan *Add other paper*.

**Add Additional Paper**

Paper presenter

Remaining paper

Select Paper \*

▼

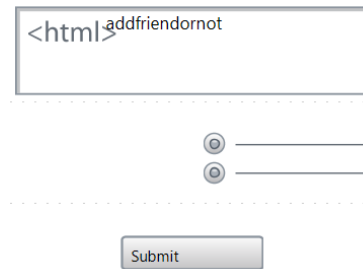
Add other paper  
☐

**SUBMIT**

**Gambar 5.27 Tampilan formulir menambahkan *paper* tambahan**

## 5. Memilih Menambah Partisipan Atau Tidak

Tahapan memilih menambah partisipan atau tidak dilakukan ketika peserta telah selesai mendaftarkan *paper* atau peserta memilih untuk tidak mendaftarkan *paper*. Gambar 5.28 di bawah ini merupakan *form* untuk memilih menambah partisipan atau tidak.



The image shows a web form interface. At the top, there is a rectangular box containing the text "<html>addfriendornot". Below this box, there are two radio button widgets, each consisting of a small circle with a dot inside and a horizontal line to its right. At the bottom of the form is a rectangular button labeled "Submit".

**Gambar 5.28 Formulir memilih menambah partisipan atau tidak**

Sama seperti aktivitas Memilih mendaftarkan *paper* atau tidak, pada aktivitas ini terdapat HTML *widget* untuk membuat tulisan perintahnya dan terdapat *radiobutton widget* untuk pilihan yang mungkin. *Radiobutton widget* ini bersifat *mandatory* sehingga peserta wajib mengisinya untuk dapat melanjutkan ke aktivitas berikutnya. Gambar 5.29 di bawah ini merupakan tampilan aktivitas Memilih menambah partisipan atau tidak.

### Participant Registration

Do you want to add participant?

\* ☐ Yes  
☐ No

SUBMIT

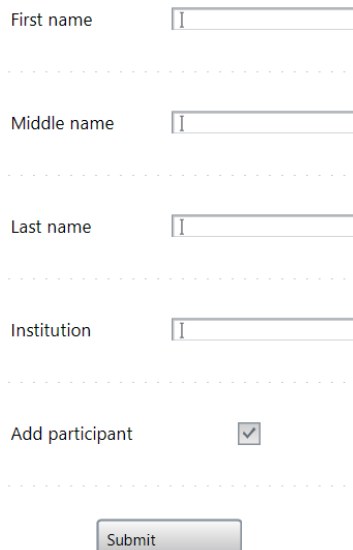
**Gambar 5.29 Tampilan formulir memilih menambah partisipan atau tidak**

## 6. Add Participant

Aktivitas *add participant* ini terjadi jika peserta pilihan *yes* pada aktivitas memilih menambah partisipan atau tidak sebelumnya. Partisipan yang dimaksud pada proses pendaftaran ini adalah peserta yang tidak mempresentasikan *paper* pada ISICO sehingga hanya mengikuti konferensi

sebagai *audience*. Gambar 5.30 merupakan *form* untuk aktivitas *add participant* ini.

Pada *form* ini, atribut *first name*, *middle name*, *last name*, dan *institution* menggunakan *field widget* dan bersifat *mandatory* kecuali *middle name*. Sedangkan *add participant* di sini menggunakan *checkbox widget* sehingga bila peserta mencentang *checkbox* tersebut, peserta akan kembali pada halaman ini untuk dapat menambahkan partisipan lagi. *Checkbox widget* ini bersifat tidak *mandatory*, sehingga peserta tidak wajib untuk mencentangnya. Selanjutnya ada tombol *submit* yang menggunakan *submit widget* untuk melakukan penyimpanan data pada *form* terkait agar dapat masuk ke aktivitas berikutnya.



First name

.....

Middle name

.....

Last name

.....

Institution

.....

Add participant ☒

.....

**Gambar 5.30** Formulir menambahkan partisipan

Gambar 5.31 merupakan tampilan aktivitas *add participant* bila proses dijalankan.

**Add Participant**

First name \*

Middle name

Last name \*

Institution \*

Add participant

☐

**SUBMIT**

**Gambar 5.31 Tampilan formulir menambahkan partisipan**

## 7. Memilih Menambah Kebutuhan Lain Atau Tidak

Tahapan memilih menambah kebutuhan lain atau tidak dilakukan ketika peserta telah selesai menambahkan partisipan atau peserta memilih untuk tidak menambahkan partisipan. Kebutuhan lain yang dimaksud dalam proses ini adalah memesan *proceeding* atau melakukan reservasi hotel. Gambar 5.32 di bawah ini merupakan *form* untuk memilih menambah kebutuhan lain atau tidak.

<html>addother

☐ \_\_\_\_\_

☐ \_\_\_\_\_

**Gambar 5.32 Formulir memilih menambah kebutuhan lain atau tidak**

Sama seperti aktivitas memilih menambah partisipan atau tidak, pada aktivitas ini terdapat *HTML widget* untuk membuat tulisan perintahnya dan terdapat *radiobutton widget* untuk pilihan yang mungkin. *Radiobutton widget* ini bersifat *mandatory* sehingga peserta wajib mengisinya untuk dapat

melanjutkan ke aktivitas berikutnya. Gambar 5.33 merupakan tampilan aktivitas memilih menambah kebutuhan lain atau tidak.

#### Memilih Menambah Kebutuhan Lain Atau Tidak

Do you want to add other?

☐ Yes

☐ No

**SUBMIT**

**Gambar 5.33 Tampilan formulir memilih menambah kebutuhan lain atau tidak**

### 8. Memesan *Proceeding*

Aktivitas memesan *proceeding* dilakukan ketika peserta memilih *yes* pada aktivitas Memilih menambah kebutuhan lain atau tidak sebelumnya. Gambar 5.34 merupakan *form* yang digunakan pada aktivitas memesan *proceeding*.

Proceeding quant...

.....

Hotel reservation ☒

.....

**Submit**

**Gambar 5.34 Formulir memesan prosiding**

Pada Gambar 5.34, *widget* yang digunakan untuk memesan *proceeding* adalah *select widget* sedangkan untuk *hotel reservation* menggunakan *checkbox widget*. Kedua *widget* tersebut tidak bersifat *mandatory* sehingga peserta tidak wajib mengisinya. Selanjutnya terdapat *submit button* agar peserta dapat lanjut ke aktivitas berikutnya. Gambar X di bawah ini merupakan tampilan aktivitas memesan *proceeding* ketika



dijalankan. Terlihat bahwa tidak ada masukan yang bersifat *mandatory*.

**Proceeding Order**

Proceeding quantity

item1 ▼

Hotel reservation

☐

**SUBMIT**

**Gambar 5.35 Tampilan formulir memesan prosiding**

## 9. Mereservasi Hotel

Aktivitas ini terjadi jika peserta mencentang *hotel reservation* pada aktivitas memesan *proceeding* sebelumnya. Aktivitas ini juga merupakan aktivitas akhir untuk pendaftaran peserta, yang kemudian akan dilanjutkan dengan aktivitas pembayaran. Pada aktivitas ini, banyak sekali masukan yang ada, yaitu *passport ID*, *full name*, *gender*, *country*, *select hotel*, *check in date*, *check out date*, *number of room*, dan *guest number*. Gambar 5.36 menggambarkan *form* mereservasi hotel.

Dari *form* pada Gambar 5.36, *widget* yang digunakan untuk *passport ID*, *full name*, *country*, *number of room*, dan *guest number* adalah *field widget*, sedangkan *gender* menggunakan *radiobutton widget*. Untuk *select hotel* digunakan *select widget*, untuk *check in* dan *check out* digunakan *date picker widget*, sedangkan untuk syarat dan ketentuan menggunakan *HTML widget*. Semua masukan pada *form* ini bersifat *mandatory* sehingga peserta harus mengisi semua *form* yang ada di aktivitas mereservasi hotel ini agar dapat menyelesaikan proses pendaftaran. Gambar 5.37 merupakan tampilan ketika aktivitas mereservasi hotel dijalankan.

Passport ID	<input type="text"/>		
Full name	<input type="text"/>	Gender	<input type="radio"/> <input type="radio"/>
Country	<input type="text"/>		
Select hotel	<input type="text"/>		
Check in	<input type="text"/>	Check out	<input type="text"/>
Number of room	<input type="text"/>	Guest number	<input type="text"/>
<div>&lt;html&gt;syarat dan ketentuan</div>			
<input type="button" value="Submit"/>			

Gambar 5.36 Formulir reservasi hotel

Book Hotel

Passport ID \*

Full name \*

Country \*

Select hotel \*

item1

Check in \*

2000/01/01 00:00:01

Check out \*

2000/01/01 00:00:01

Number of room \*

Guest number \*

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

SUBMIT

Gambar 5.37 Tampilan mereservasi hotel

## 10. Mengunggah Bukti Pembayaran

Tahap selanjutnya setelah pendaftaran selesai adalah pembayaran. Tahapan ini dilakukan pada aktivitas mengunggah bukti pembayaran. Gambar 5.38 merupakan *form* yang digunakan pada aktivitas mengunggah bukti pembayaran ini.

Pada Gambar 5.38 *invoice* ID yang menggunakan *field widget* ini bersifat *read only* dan diisi oleh panitia penyelenggara ISICO. Untuk *invoice* dan *payment receipts* menggunakan *file widget* karena di sini disertakan *attachment* yang diambil di luar sistem. Namun untuk *invoice* sendiri, *attachment* didapatkan dari aktivitas yang dilakukan oleh panitia penyelenggara, sehingga peserta hanya harus menyertakan *attachment* pada *payment receipts* dengan mengunggah *file* bukti pembayaran.

Invoice id

.....

Invoice

.....

Payment Receipts

.....

**Gambar 5.38** Formulir mengunggah bukti pembayaran

Untuk konfigurasi *file widget* dilakukan pada *tab General* kemudian masuk ke dalam *tab Data*. Gambar 5.39 merupakan tampilan saat konfigurasi *file widget*.



**Data**

☐ Download only   ☐ Displays a preview if the document/resource is an image.

Initial value ☒ Document [i](#)   ☐ Resource [i](#)

File widget type ☐ Text field, for URL   ☒ Button, to browse for file   ☐ Both, with radio buttons for user to choose

Field modifier [i](#)

Output operation   [Set document](#) [i](#)  

**Gambar 5.39 Konfigurasi *file widget***

Di sini, variabel yang digunakan untuk *file widget* dibuat saat *file widget* ini ada dengan memilih *create a document*. Setelah itu, konfigurasi dilanjutkan dengan mengisikan *set document* dengan nama *file widget* yang terdapat pada *form* terkait. Gambar 5.40 merupakan tampilan *form* mengunggah bukti pembayaran saat dijalankan.

**Mengupload Bukti Pembayaran**

Invoice id

Invoice

☒ URL ☐ File

Payment Receipts \*

**Gambar 5.40 Tampilan formulir mengunggah bukti pembayaran****11. Melakukan Konfirmasi Kehadiran**

Aktivitas ini dilakukan ketika pihak panitia penyelenggara telah mengirimkan bukti pelunasan pembayaran (*receipt*) dan *qrcode* kepada peserta. Gambar 5.41 merupakan *form* yang digunakan pada aktivitas melakukan konfirmasi kehadiran ini.

Pada *form* ini, terdapat *file widget* yang digunakan pada *payment receipt* dan *qrcode*. Kedua *widget* tersebut berfungsi menampilkan masukan berupa *receipt* dan *qrcode* yang dikirimkan oleh panitia penyelenggara. Kemudian terdapat

HTML *widget* serta *radiobutton widget* yang digunakan untuk memilih konfirmasi kehadiran.

The form consists of two HTML widgets. The first widget is labeled "qrcode & payment receipts" and contains a text input field. The second widget is labeled "konfirmasikehadiran" and contains a text input field. Below the first widget is a label "Payment receipts" followed by a text input field. Below the second widget is a label "Qrcode" followed by a text input field. At the bottom of the form is a "Submit" button.

**Gambar 5.41 Formulir konfirmasi kehadiran**

Untuk pemilihan konfirmasi ini bersifat *mandatory*, sehingga peserta wajib mengisinya. Gambar 5.42 merupakan tampilan *form* melakukan konfirmasi kehadiran saat proses dijalankan.

### Melakukan Konfirmasi Kehadiran

The form is titled "Melakukan Konfirmasi Kehadiran". It starts with the text "Here are your qrcode and payment receipts". Below this are two sections: "Payment receipts" and "Qrcode". Each section has two radio buttons: "URL" (selected) and "File". Below each section is a text input field. Below these is the question "Will you attend the conference?" with three radio button options: "item1" (selected), "item2", and "item3". At the bottom is a "SUBMIT" button.

**Gambar 5.42 Tampilan formulir konfirmasi kehadiran**

## 12. Melakukan Konfirmasi Kebenaran Nama

Tahapan melakukan konfirmasi kebenaran nama dilakukan ketika peserta memilih *yes* saat konfirmasi kehadiran. Gambar 5.43 di bawah ini merupakan *form* untuk melakukan konfirmasi kebenaran nama.

Pada Gambar 5.43 menunjukkan bahwa aktivitas ini memiliki *HTML widget* untuk membuat tulisan perintahnya dan terdapat *radiobutton widget* untuk pilihan yang mungkin. *Radiobutton widget* ini bersifat *mandatory* sehingga peserta wajib mengisinya untuk dapat melanjutkan ke aktivitas berikutnya. Gambar 5.44 di bawah ini merupakan tampilan aktivitas melakukan konfirmasi kebenaran nama.

**Gambar 5.43 Formulir konfirmasi kebenaran nama**

Terlihat bahwa pada Gambar 5.44, peserta harus memilih salah satu dari pilihan yang ada untuk dapat melanjutkan ke aktivitas selanjutnya atau mengakhiri proses pendaftaran ISICO yang dilakukan. Jika peserta memilih *yes*, peserta akan mengakhiri seluruh proses pendaftaran ISICO karena semua aktivitas telah dilakukan. Namun jika memilih *no*, peserta akan melakukan pembenaran nama dengan mengisi *form* pembenaran nama yang sudah disediakan.

## 13. Mengisi *Form* Pembenaran Nama

Tahapan ini dilakukan ketika peserta memilih *yes* pada aktivitas sebelumnya, yaitu melakukan konfirmasi kebenaran

nama. Gambar 5.45 merupakan *form* yang digunakan dalam aktivitas mengisi *form* kebenaran nama.

### Melakukan Konfirmasi Kebenaran Nama

Is/are the names right?

- \* ☐ Yes  
☐ No

SUBMIT

**Gambar 5.44 Tampilan formulir kebenaran nama**

Dari Gambar 5.45 di bawah ini, *field ID*, *first name*, *middle name*, *last name* menggunakan *field widget*. Peserta harus mengisi ID *paper* pada *field ID* bila jenis yang dipilih merupakan *presenter* yang tercantum pada *invoice*. Sedangkan *participant*, peserta harus mengisi *user ID* pada *field ID*. Untuk *field first name*, *middle name*, dan *last name* diisi dengan nama yang dibenarkan. Selanjutnya terdapat *select widget* yang digunakan pada *type*, dan *checkbox widget* yang digunakan pada *justify again* untuk dapat kembali pada aktivitas ini bila dicentang.

ID	<input type="text"/>
.....	
First name	<input type="text"/>
.....	
Middle name	<input type="text"/>
.....	
Last name	<input type="text"/>

Type

Justify again ☒

**Gambar 5.45 Formulir pembetulan nama**

Gambar 5.46 merupakan tampilan bila *form* aktivitas mengisi *form* pembetulan nama dijalankan. Terlihat pada tampilan ini bahwa ada beberapa *field* yang bersifat *mandatory*, yaitu ID, *first name*, *last name*, dan *type*.

**Mengisi Form Pembetulan Nama**

ID \*

First name \*

Middle name

Last name \*

Type \*

Justify again ☐

**Gambar 5.46 Tampilan formulir pembetulan nama**

#### 5.4.2. Formulir Bagian Keuangan

Formulir bagian keuangan merupakan formulir yang digunakan untuk pihak penyelenggara bagian keuangan dalam melakukan pengiriman *invoice*, *receipt*, dan *qrcode*. Berikut ini adalah aktivitas-aktivitas yang dilakukan oleh bagian keuangan.

##### 1. Mengirimkan *Invoice*



Tahapan mengirimkan *invoice* ini dilakukan ketika peserta telah menyelesaikan tahapan pendaftaran dan sebelum peserta melakukan pengunggahan bukti pembayaran. Gambar 5.47 di bawah ini merupakan *form* yang digunakan untuk aktivitas mengirimkan *invoice*.

Pada Gambar 5.47, *registrant ID* dan *email registrant* menggunakan *field widget* dan bersifat *read only* karena kedua *field* tersebut untuk menampilkan informasi *registrant* (peserta) yang telah menyelesaikan pendaftaran. Untuk *invoice ID* menggunakan *field widget* dan bersifat *mandatory*, sedangkan *invoice* menggunakan *file widget* dan juga bersifat *mandatory*. *File invoice* di sini dibuat dari sistem eksternal (pembuatan tidak menggunakan aplikasi Bonita BPM).

Registrant ID

.....

Email Registrant

.....

Invoice ID

.....

Invoice

.....

**Gambar 5.47** Formulir mengirimkan *invoice*

Gambar 5.48 merupakan tampilan *form* ketika proses dijalankan. Terlihat bahwa *invoice ID* dan *invoice* bersifat *mandatory* yang berarti bagian keuangan harus mengisinya agar dapat melanjutkan aktivitas berikutnya.

**Mengirimkan Invoice**

Registrant ID

Email Registrant

Invoice ID \*

Invoice \*

**Gambar 5.48 Tampilan formulir mengirimkan *invoice***

## 2. Meverifikasi Bukti Pembayaran

Tahapan ini dilakukan setelah peserta telah menyelesaikan aktivitas mengunggah bukti pembayaran. Gambar 5.49 di bawah ini merupakan tampilan *form* yang digunakan pada aktivitas meverifikasi bukti pembayaran.

Invoice id

.....

Payment receipt

.....

Note

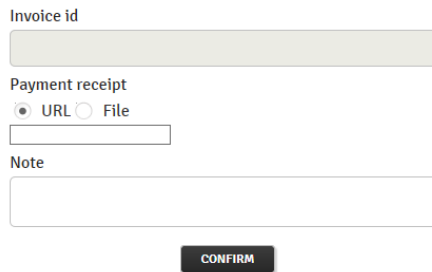
.....

**Gambar 5.49 Formulir verifikasi bukti pembayaran**

Pada Gambar 5.49 di atas, terdapat *field invoice* ID yang bersifat *read only*, karena menggunakan variabel hasil dari aktivitas sebelumnya, yaitu mengunggah bukti pembayaran yang dilakukan oleh peserta. Lalu ada *payment receipt* yang

merupakan *file widget* dimana juga menggunakan variabel hasil dari aktivitas sebelumnya. Terdapat *note* yang menggunakan *text area widget*, berfungsi untuk menambahkan keterangan terkait konfirmasi bukti pembayaran ini. Gambar 5.50 merupakan tampilan *form* meverifikasi bukti pembayaran bila sistem dijalankan.

#### **Meverifikasi Bukti Pembayaran**



Invoice id

Payment receipt

☒ URL ☐ File

Note

CONFIRM

**Gambar 5.50 Tampilan formulir verifikasi bukti pembayaran**

### **3. Mengirim Tanda Bukti Pembayaran dan *Qrcode***

Tahapan mengirim tanda bukti pembayaran dan *qrcode* merupakan tahapan bila pihak penyelenggara telah melakukan konfirmasi pembayaran. Tanda bukti pembayaran di sini merupakan tanda bukti bahwa pembayaran yang dilakukan oleh peserta telah lunas. Gambar 5.51 merupakan *form* yang digunakan untuk aktivitas mengirim tanda bukti pembayaran dan *qrcode*.

Payment Receipt

QR Code

Send

**Gambar 5.51 Formulir mengirim bukti penerimaan pembayaran dan *qrcode***

Dari Gambar 5.51 di atas, terlihat bahwa kedua *field* tersebut menggunakan *file widget* dan keduanya bersifat *mandatory* sehingga bagian keuangan wajib mengisi *field* tersebut agar bisa lanjut ke aktivitas berikutnya. Gambar 5.52 di bawah ini merupakan tampilan *form* mengirim tanda bukti pembayaran dan *qrcode* bila proses dijalankan.

#### Mengirim Tanda Bukti Pembayaran dan Qrcode

Payment Receipt \*  Browse...

QR Code \*  Browse...

SEND

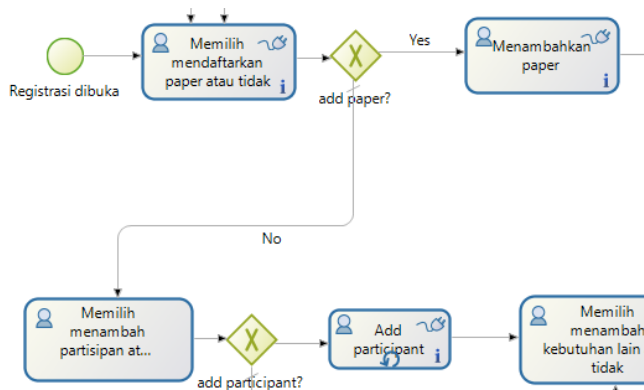
**Gambar 5.52 Tampilan formulir mengirim bukti penerimaan pembayaran dan *qrcode***

## 5.5. Konfigurasi Fungsi *Gateway*

*Gateway* dalam aplikasi Bonita BPM berfungsi untuk mengatur jalan dari proses jika memiliki lebih dari satu kemungkinan atau biasa disebut percabangan. Pada kasus proses pendaftaran ISICO ini, *gateway* yang digunakan adalah yang memiliki tipe XOR dikarenakan di setiap percabangan yang ada, harus ada satu jalan yang diproses setelah keluar dari *gateway*. Oleh karena yang dijalankan tepat satu proses, maka *gateway* ini perlu dikonfigurasi. Gambar 5.53 di bawah

ini merupakan contoh penggunaan *connector* dengan tipe XOR.

Pada Gambar 5.53, setelah peserta melakukan aktivitas memilih mendaftarkan *paper* atau tidak, proses akan memasuki *gateway* bertipe XOR, yang berarti proses selanjutnya adalah menambahkan *paper* jika peserta memilih *yes* atau memilih menambah partisipan atau tidak jika peserta memilih *no*. Cara melakukan konfigurasi adalah pertama pengguna harus menentukan jalur *default* terlebih dahulu. Pada kasus di atas, jalur *default* adalah jalur *no* yang ditandai dengan adanya garis miring pada *line no*. Untuk mengaturnya, klik pada *line* yang diinginkan, bisa *no* atau *yes*, lalu masuk pada *tab General* dan centang pernyataan *Default flow* seperti pada Gambar 5.54.

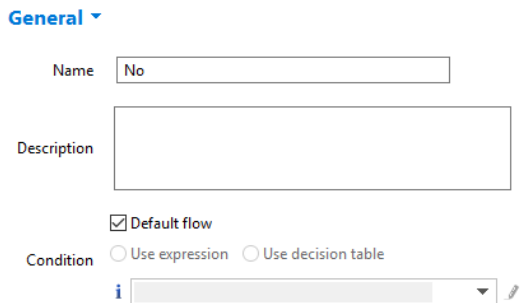


**Gambar 5.53 Penggunaan *connector gateway* bertipe XOR**

Selanjutnya, *line* yang lain juga harus dikonfigurasi dengan memasukkan kondisi tidak *default*. Gambar 5.55 merupakan contoh dari penentuan kondisi pada *line* yang tidak *default*.

Dari Gambar 5.55 di bawah ini, diketahui bahwa *line yes* akan dilalui ketika variabel *is\_addPaper* bernilai *yes*, yang berarti jika peserta memilih *yes* saat berada di aktivitas memilih

mendaftarkan *paper* atau tidak, maka peserta akan diteruskan pada aktivitas yang dilalui *line yes*, yaitu menambahkan *paper*.



**General** ▾

Name

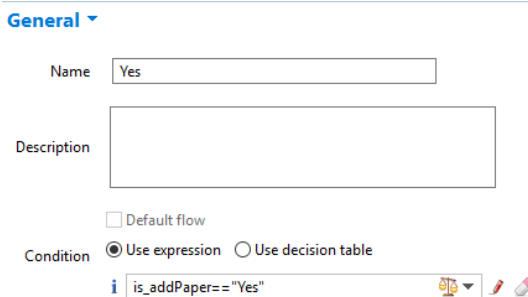
Description

☒ Default flow

Condition ☐ Use expression ☐ Use decision table

*i*

**Gambar 5.54** Konfigurasi *default flow* pada XOR gateway



**General** ▾

Name

Description

☐ Default flow

Condition ☒ Use expression ☐ Use decision table

*i*

**Gambar 5.55** Konfigurasi penentuan kondisi pada XOR gateway

## 5.6. Konfigurasi Fungsi *Loop*

Fungsi *loop* pada Bontia BPM digunakan untuk melakukan pengulangan suatu aktivitas jika memenuhi kondisi tertentu. Pada proses pendaftaran ISICO ini terdapat beberapa aktivitas yang menggunakan fungsi *loop*, seperti menambahkan *paper* tambahan (*additional paper*). Hal ini dikarenakan, saat peserta mendaftarkan *additional paper*, peserta diperbolehkan mendaftarkan lebih dari satu *additional paper*. Gambar 5.56 di bawah ini merupakan cara untuk melakukan konfigurasi fungsi *loop* pada aktivitas menambahkan *paper* tambahan.

**Iteration** ▾

☐ None 
 ☒ Standard Loop 
 ☐ Parallel mu

Only process scope variables can be used in the loop condition

☒ Test after 
 ☐ Test before

Loop while

Maximum loop

**Gambar 5.56 Konfigurasi fungsi *loop***

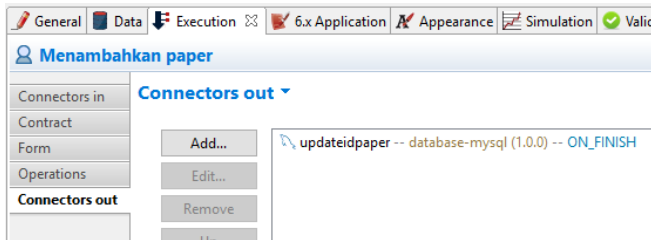
Dari Gambar 5.56 tersebut, penulis memilih *standard loop* dan memilih konfigurasi *test after* sehingga pengujian untuk kondisinya akan selesai setelah iterasi berakhir. Kondisi yang diperlukan, yang berupa variabel bertipe *boolean*, dimasukkan ke dalam *loop while* sehingga pada Gambar 5.56 menunjukkan bahwa aktivitas *add paper* akan melakukan iterasi bila kondisi *is\_addAdditionalPaper* terpenuhi. Kemudian, angka pada *maximum loop* dimaksudkan sebagai jumlah maksimal iterasi dilakukan.

## 5.7. Konfigurasi *Connector*

*Connector* merupakan ekstensi dari aplikasi Bonita BPM yang memungkinkan Bonita terhubung dengan aplikasi luar, seperti DBMS eksternal, *email*, ERP, dan lain-lain. Pada penelitian ini, penulis menggunakan *email connector* sebagai pemberi notifikasi saat berpindah *lane* dari peserta ke bagian keuangan atau sebaliknya. Selain itu, penulis juga menggunakan DBMS *connector*, tepatnya MySQL, yang digunakan untuk menyimpan data pendaftaran sehingga dapat terhubung oleh sistem eksternal.

Gambar 5.57 di bawah menggambarkan cara konfigurasi *connector* dalam Bonita BPM. Langkah pertama yang dilakukan adalah memilih aktivitas yang akan diberi *connector*, lalu masuk ke *tab execution* dan memilih antara *connectors in* atau *connectors out*. Perbedaannya adalah jika

*connectors in*, maka *connector* akan dieksekusi sebelum memasuki aktivitas yang dipilih. Sebaliknya, jika *connectors out*, maka *connector* akan dieksekusi setelah menjalankan aktivitas yang dipilih.

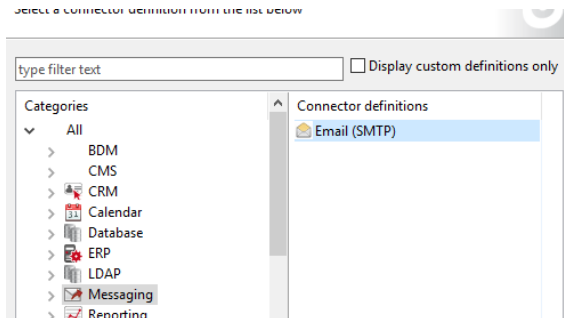


**Gambar 5.57 Konfigurasi *connector***

Jika telah menentukan *connectors in* atau *out*, maka selanjutnya tekan tombol *add*, lalu masukkan nama *connector* dan tentukan jenis *connector* yang ingin dibuat.

#### 5.7.1. *Connector Email*

Untuk membuat *connector email*, yang dipilih adalah *connector* berjenis *messaging* dan pilih *email* (SMTP) seperti Gambar 5.58.



**Gambar 5.58 Memilih *connector messaging***

Selanjutnya adalah memasukkan *username* dan *password email* yang ingin digunakan untuk mengirimkan *email* seperti gambar 5.59.



**Email (SMTP) (1.0.0)**

**Connection Information**  
Enter the connection SMTP server connection information

SMTP Host \*

SMTP Port \*

► Security

▼ Authentication

Username

Password

**Gambar 5.59 Konfigurasi informasi koneksi email**

Setelah itu memasukkan pengirim dan penerima pesan. Untuk penerima, diambil dari data peserta yang tersimpan dalam variabel seperti yang ada pada gambar 5.60 di bawah ini.

**Email (SMTP) (1.0.0)**

**Email addressee**  
Configure the email addressees and sender

From \*

To \*

► Other

**Gambar 5.60 Konfigurasi alamat email**

Kemudian menentukan *subject message* dan *attachment* apabila diperlukan. Untuk penulisan *message* di sini menggunakan bahasa HTML jika ingin mengambil variabel proses. Gambar 5.61 merupakan tampilan konfigurasi konten email.

Setelah melakukan konfigurasi dalam Bonita BPM, selanjutnya adalah melakukan konfigurasi pada email yang dimasukkan saat *authentication* (Gambar 5.59). Caranya adalah masuk ke email tersebut dan masuk ke menu *setting* kemudian masuk ke *tab forwarding and POP/IMAP*. Kemudian aktifkan *POP download* dan *IMAP access*, dengan mencentang pada *checkbox* yang tersedia.

**Email (SMTP) (1.0.0)**

**Email message**  
Set the email content

Subject \* Registration Success

Message  
Your registration is successfully submitted.  
Please wait for 2x24 to receive an invoice.  
Regards,  
ISICO

[Switch editor...](#)

☒ Use HTML  
[Switch editor to create a condition...](#)

▼ **Attachments**  
List of attachments

Add row Remove row

[Edit as an expression](#)

**Gambar 5.61 Konfigurasi konten email**

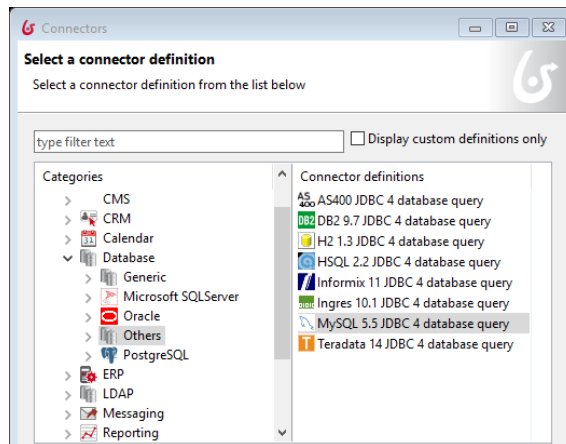
Centang semua menu yang ada di halaman tersebut, terutama yang berkaitan dengan pengiriman *email* yang melibatkan lingkungan eksternal. Gambar 5.62 berisikan semua informasi yang dijelaskan sebelumnya.

<b>Penerusan:</b> <a href="#">Pelajari selengkapnya</a>	<span>Tambahkan alamat penerusan</span>  Kiat: Anda juga dapat meneruskan sebagian email dengan cara <a href="#">membuat filter!</a>
<b>Download via POP:</b> <a href="#">Pelajari selengkapnya</a>	1. <b>Status: POP diaktifkan</b> untuk semua email yang masuk sejak 22/06/12 <input checked="" type="radio"/> Aktifkan POP untuk <b>semua email</b> (termasuk email yang telah didownload) <input type="radio"/> Aktifkan POP untuk <b>email yang tiba dari sekarang</b> <input type="radio"/> <b>Nonaktifkan POP</b>  2. <b>Ketika pesan diakses dengan POP</b> <span>simpan salinan Gmail di Kotak Masuk</span>  3. <b>Konfigurasi program email Anda</b> (misal: Outlook, Eudora, Netscape Mail) <a href="#">Petunjuk konfigurasi</a>
<b>Akses IMAP:</b> <small>(Jika Gmail dari aplikasi lain menggunakan IMAP)</small> <a href="#">Pelajari selengkapnya</a>	<b>Status: IMAP dinonaktifkan</b> <input checked="" type="radio"/> Aktifkan IMAP <input type="radio"/> Nonaktifkan IMAP  <b>Bila saya menandai pesan di IMAP sebagai dihapus:</b> <input checked="" type="radio"/> Buang Otomatis aktif - Segera perbarui server. (default) <input type="radio"/> Buang Otomatis nonaktif - Tunggu program memperbarui server.  <b>Bila pesan ditandai sebagai dihapus dan dibuang dari folder IMAP yang terakhir ditampilkan:</b> <input checked="" type="radio"/> Arsipkan pesan (default) <input type="radio"/> Pindahkan pesan ke Kotak Sampah <input type="radio"/> Segera hapus pesan selamanya

**Gambar 5.62 Konfigurasi SMTP email**

### 5.7.2. Connector MySQL

Di dalam Bonita BPM, interaksi dengan MySQL yang digunakan dalam penelitian ini meliputi *read*, *insert*, dan *update*. Untuk membuat *connector* MySQL, pilih kategori *Database* lalu MySQL 5.5 JDBC 4 *database query* seperti pada Gambar 5.63 di bawah ini.

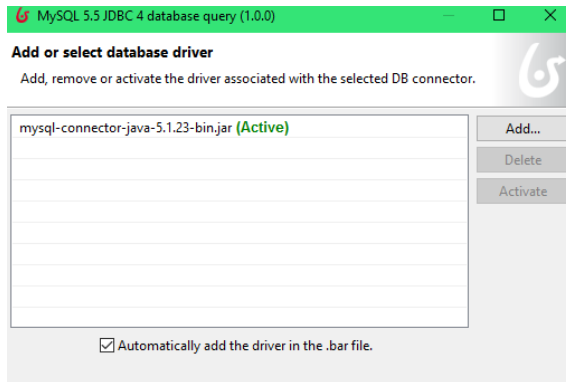


**Gambar 5.63** Memilih jenis *connector database* MySQL

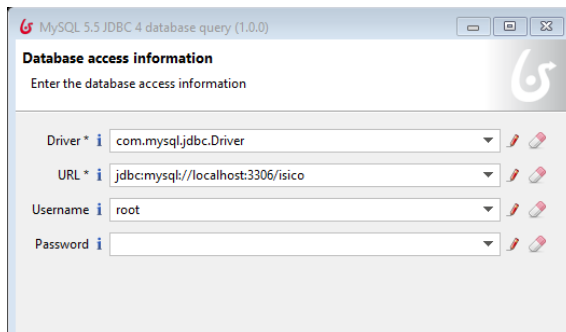
Setelah itu harus mengisi nama *connector* dan menambahkan *driver*. Di dalam Bonita sendiri sudah terdapat *driver* yang dapat digunakan untuk menghubungkan Bonita dengan MySQL seperti yang terlihat pada Gambar 5.64.

Kemudian memasukkan informasi untuk mengakses *database*, seperti *driver*, URL, *username*, dan *password*. Pada 5.65 di bawah ini terlihat bahwa *database* yang dipakai memiliki URL *localhost:3306/isico*, dimana *isico* ini merupakan nama *database*.

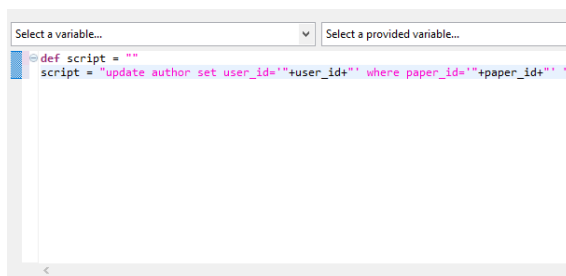
Selanjutnya adalah memasukkan *query* yang diperlukan pada aktivitas tersebut dalam sebuah *editor* seperti yang terlihat pada Gambar 5.66.



**Gambar 5.64 Konfigurasi *driver database***



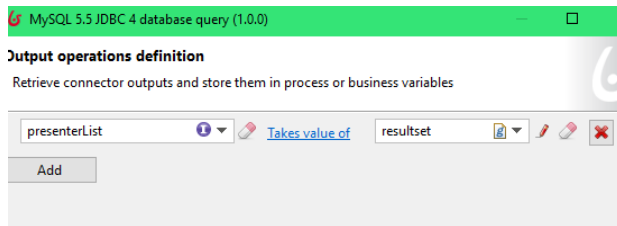
**Gambar 5.65 Konfigurasi informasi pengaksesan *database***



**Gambar 5.66 Editor untuk memasukkan *query***

Konfigurasi *connector* ke MySQL untuk proses *insert* dan *update* cukup berhenti sampai di sini. Namun, untuk

konfigurasi untuk menampilkan data dari *database* yang menggunakan *query select*, maka harus dilanjutkan pada konfigurasi berikutnya, dimana data yang telah diambil dari *database* MySQL disimpan ke dalam variabel proses yang ada di dalam Bonita BPM. Caranya adalah setelah dari *editor*, lanjut ke tampilan *output operations definition* dan pilih *scripting mode* untuk membuat *array* yang menyimpan data yang telah diambil dari MySQL. Kemudian dilanjutkan lagi dengan mendefinisikan variabel yang digunakan untuk penyimpanan ini seperti yang terlihat pada Gambar 5.67 di bawah ini.



**Gambar 5.67 Konfigurasi keluaran operasi *select database***

Sebagai catatan, bahwa variabel yang digunakan untuk menyimpan data yang telah diambil dari *database* harus bertipe Java Object, dan untuk menyimpan data dalam jumlah banyak (*list*) seperti contoh pada Gambar 5.67 di atas untuk variabel *presenterList*, menggunakan tipe Java Util List. Kemudian memasukkan *script* penyimpanan pada *field takes value of* dengan menekan ikon pensil.

```

1. List result = new ArrayList();
2. While (resultset.next()) {
3.   Result.add(resultset.getString(1))
4. }
5. Return result;

```

**Gambar 5.68 Script menyimpan data ke variabel proses**

Gambar 5.68 di atas merupakan kode untuk menyimpan data yang diambil dari *database* ke dalam variabel proses dengan *array*. Baris pertama pada kode tersebut berfungsi untuk membuat *array list* baru. Baris kedua dan ketiga adalah untuk menambahkan data ke dalam *array list* yang telah dibuat pada baris pertama, dan baris kelima merupakan fungsi untuk mengembalikan nilai atau hasil *array*.

### 5.8. Publish Organization

Proses *publishing* dilakukan ketika proses yang telah dibuat siap untuk dijalankan. Cara melakukan *publishing* adalah dengan masuk ke *menu organization* pada *menu bar* lalu pilih *publish*. Kemudian pilih organisasi yang akan digunakan lalu tekan tombol *next* jika ingin melakukan konfigurasi lanjutan berupa menambahkan *default* pengguna yang akan langsung masuk ke dalam proses saat proses dijalankan. Gambar 5.69 menunjukkan tampilan ketika menentukan pengguna *default*.

**Default user**  
Select the username that will be used to log in to Portal

Username	delinare
----------	----------

**Gambar 5.69** Konfigurasi pengguna *default* sistem

Jika sudah, tekan tombol *publish* untuk melakukan *publishing*. Bila berhasil, organisasi yang telah *publish* memiliki tanda yang bertuliskan *active* seperti pada Gambar 5.70 di bawah ini.

Upload to local Portal...	
Upload to local Portal...	
Load the selected organization into the local Bonita BPM Portal	
Name	Description
ACME	The ACME organization is an example of a typical hierarch...
ISICO (active)	
Pemesanan Tiket Online	
Proses Review Makalah	

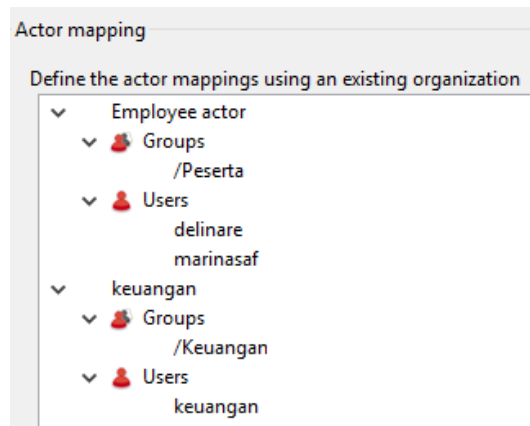
**Gambar 5.70** Publish organization

## 5.9. Deklarasi Organisasi

Sebelum menjalankan proses, langkah selanjutnya adalah melakukan deklarasi organisasi. Kegiatan ini bertujuan untuk mendeklarasikan atau menjelaskan pemetaan yang ada dalam proses Bonita BPM. Cara melakukannya adalah dengan masuk ke *icon configure*. Berikut ini adalah tahapan yang perlu dilakukan saat deklarasi organisasi.

### 5.9.1. Konfigurasi Actor Mapping

Konfigurasi *actor mapping* ini bertujuan untuk melakukan spesifikasi terhadap aktivitas yang dapat dilakukan untuk setiap pengguna. Caranya dengan menambahkan dan mencocokkan antara pengguna, *group*, dan *role*. Gambar 5.71 di bawah ini merupakan gambaran pemetaan aktor yang dilakukan pada penelitian ini.

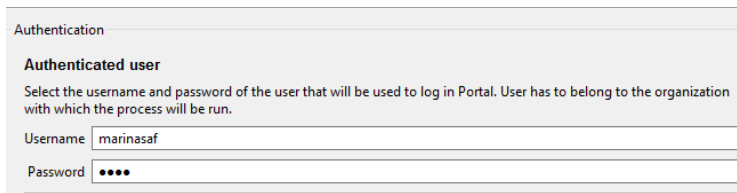


Gambar 5.71 Konfigurasi actor mapping

### 5.9.2. Konfigurasi Autentikasi

Konfigurasi autentikasi bertujuan untuk mempermudah saat proses dijalankan, karena pengguna yang memiliki nama dan *password* yang dimasukkan di sini akan langsung *logged in* saat proses dijalankan. Pengguna yang dimaksud di sini

haruslah pengguna yang sama saat *publish organization* sebelumnya. Akan tetapi, konfigurasi ini juga dapat dikosongkan sehingga pengguna menginginkan untuk tidak langsung masuk ke dalam proses saat dijalankan. Hal ini memungkinkan untuk *login* ke sistem dengan pengguna yang berbeda. Gambar 5.72 di bawah ini merupakan tampilan untuk konfigurasi autentikasi.



Authentication

**Authenticated user**

Select the username and password of the user that will be used to log in Portal. User has to belong to the organization with which the process will be run.

Username

Password

**Gambar 5.72 Konfigurasi autentikasi pengguna**

## 5.10. Sistem Eksternal

Pada tahapan ini dilakukan implementasi terhadap sistem eksternal yang dibangun untuk memenuhi kebutuhan pengguna yang tidak dibangun dalam Bonita BPM. Kebutuhan yang dibangun dalam sistem eksternal adalah fitur pendaftaran akun Bonita, pembuatan *invoice*, pembuatan *qrcode*, serta informasi *dashboard* yang memuat informasi jumlah *author* yang terdaftar, jumlah *participant (non author)* yang terdaftar, jumlah peserta yang telah melunasi pembayaran, dan jumlah peserta yang telah hadir dalam seminar.

### 5.10.1. Implementasi Fitur Pendaftaran Akun Bonita

Implementasi fitur pendaftaran akun bonita di sini bertujuan sebagai pengajuan akun yang nantinya digunakan untuk masuk ke dalam sistem yang ada di Bonita BPM. Pengguna harus mengisi *email*, *username*, *password*, *first name*, *middle name*, *last name*, *country*, *institution*, dan *address*. Gambar 5.73 merupakan tampilan fitur pendaftaran akun bonita. Potongan kode untuk fitur pendaftaran akun Bonita dapat dilihat pada Gambar 5.74 di bawah ini.



**Gambar 5.73 Tampilan fitur pendaftaran akun Bonita**

```

1.  function index(){
2.    $this->load->model('model_registrant');
3.    $judulpage = "Daftar registrant";
4.    $data['judulpage'] = $judulpage;
5.    $data['registrant'] = $this->
      model_registrant->list_registrant()-
      >result();
6.    $this->load->view('registrant',$data);
7.  }

```

**Gambar 5.74 Potongan kode pada *controllers* fitur pendaftaran akun Bonita**

Dari kode pada Gambar 5.74 di atas, baris kedua menunjukkan bahwa fungsi ini mengambil *model registrant*. Baris ketiga menunjukkan bahwa fungsi ini membuat judul halaman yang

kemudian disimpan ke dalam *array* pada baris ke-4. Baris ke-5 menyimpan hasil *query* pada *model registrant*. Kemudian fungsi ini menampilkan tampilan bernama *registrant* dengan data dalam *array data*.

```

1.  <?php                                     echo
    form_open('registrant/input_simpan'); ?>
2.  <div class="form-group has-feedback">
3.  <?php  echo  form_input('user_email','',
    array('type'=>"email",
    'placeholder'=>"Email",      'class'=>"form-
    control")); ?>
4.  <span class="glyphicon glyphicon-envelope
    form-control-feedback"></span>
5.  </div>
6.  <?php                                     echo
    form_submit('submit','Register',
    array('class'=>"btn btn-primary btn-block
    btn-flat")); ?>

```

**Gambar 5.75 Potongan kode pada *model* fitur pendaftaran akun Bonita**

Gambar 5.75 di atas merupakan potongan kode yang digunakan pada tampilan fitur pendaftaran akun Bonita. Baris pertama menunjukkan *syntax* pembuatan *form* pada *framework codeigniter*. Baris pertama tersebut menandakan bahwa aksi *form* setelah pengguna menekan tombol *submit*, maka fungsi akan menuju ke *controllers registrant* dengan fungsi *input\_simpan*. Baris kedua menunjukkan pembuatan *div* dengan *class* yang disebutkan pada baris kode. Baris ketiga merupakan *syntax input form* yang terdapat dalam *framework codeigniter*. Pada baris ini dapat ditambahkan sesuai dengan jumlah *field* yang diinginkan. Baris keempat berfungsi untuk memperbaiki tampilan *form* dengan menambahkan *icon* pada masing-masing *field*. Baris ke-6 merupakan *syntax submit*

*form* yang terdapat pada *framework codeigniter*. Selanjutnya *form* akan ditutup oleh baris ke-7.

```

1.  function input_simpan(){
2.  $dataregistrant = array(
3.  'user_email'      =>      $this->input-
    >post('user_email'),
4.  );
5.  $this->db-
    >insert('registrant',$dataregistrant);
6.  }

```

**Gambar 5.76** Potongan kode *controller* penyimpanan data registrasi akun Bonita

Gambar 5.76 di atas merupakan potongan kode pada *controller* dengan fungsi *input\_simpan*. Baris ke 2-4 menunjukkan cara pengambilan masukan pada *form* registrasi, yaitu dengan menyimpannya ke dalam *array*. Kemudian baris kelima menunjukkan cara penyimpanan data yang telah disimpan dalam *array* tadi ke dalam *database*.

#### 5.10.2. Implementasi Fitur Pembuatan Invoice

Implementasi fitur pembuatan *invoice* ini bertujuan untuk memenuhi kebutuhan penyelenggara ISICO dalam hal pembuatan *invoice*.

Gambar 5.77 merupakan tampilan fitur pembuatan *invoice* pada halaman *view invoice*. Potongan kode untuk *list invoice* pada bagian *controller* adalah yang terlihat pada Gambar 5.78.

Pada potongan kode pada Gambar 5.78, sistem akan menyimpan semua data pada variabel *array*. Baris kedua mendefinisikan judul halaman yang kemudian disimpan pada *array* di baris ketiga. Baris keempat menunjukkan bahwa sistem akan mengambil model *invoice* yang akan digunakan pada fungsi *view* ini.

📄 View Invoice				
<a href="#">Back</a>   <a href="#">View invoice</a>   <a href="#">Save as Pdf</a>				
Bill to Marina Safitri				
Pandegling				
Invoice no. 201703298896B2				
Paper Registration				
Paper ID	Paper Title	Presenter	Type	Price
5	Web Tags Formatting with Multilevel Numbering	Marius Limpo	Main paper	4,000,000
4	Usability Testing on Flight Searching Website Using Heuristic Evaluation	Marius Limpo	Additional paper	2,000,000
Participant Registration				
Name	Institution	Price		
Stezar Priansya	ITS	200,000		

**Gambar 5.77 Tampilan fitur pembuatan *invoice***

```

1. Function view() {
2.   $judulpage = "View invoice";
3.   $data['judulpage'] = $judulpage;
4.   $this->load->model('model_invoice');
5.   $user_id = $this->uri->segment(3);
6.   $data['invoice'] = $this->model_invoice->
    >get_invoice($user_id)->result();
7.   $data['invoice_proceeding'] = $this->
    >model_invoice->get_proceeding($user_id)->
    >result();
8.   if (strtotime($data['query_date'])[0]->
    >timestamp) < strtotime("2017-03-08
    14:27:03"))

```

```

9.  {
10. $data['query_paper']          =          $this-
    >model_invoice->get_paper($user_id)-
    >result();
11. $data['query_price']          =          $this-
    >model_invoice->get_price_eb($user_id)-
    >result();
12. }
13. else {
14. $data['query_price']          =          $this-
    >model_invoice->get_price($user_id)-
    >result();
15. $data['query_paper']          =          $this-
    >model_invoice->get_paper_ne($user_id)-
    >result();
16. }
17. $this->load->view('invoice_view',$data);
18. }

```

**Gambar 5.78 Potongan kode pada *controllers* fitur pembuatan *invoice***

Baris kelima menunjukkan bahwa sistem menyimpan segmen ketiga pada URL dalam sebuah variabel *user\_id*. Baris ke-6 sampai baris ke-7 menunjukkan contoh bahwa sistem mengambil *query* yang tersimpan pada *model invoice*. Gambar 5.79 merupakan contoh *query* pada *model invoice* yang diambil pada baris keenam. *Query* disimpan dalam variabel *invoice* pada baris ketiga lalu nilainya dikembalikan pada baris keempat.

Kembali pada Gambar 5.78, pada baris ke-16, sistem akan melakukan fungsi jika-maka. Hal ini ditujukan untuk melakukan perhitungan *earlybird* dengan menggunakan *timestamp* yang tersimpan pada tabel *invoice* dalam basis data.

```

1. function list_invoice()
2. {
3.     $invoice = $this->db->get('invoice');
4.     return $invoice;
5. }

```

**Gambar 5.79** Potongan kode pada *model* fitur pembuatan *invoice*

Dari fungsi pada baris ke-16 tersebut menunjukkan bahwa jika *timestamp* pada *invoice* kurang dari “2017-03-08 14:27:03”, maka sistem akan mengambil harga *earlybird* yang ditunjukkan pada baris 18-20. Namun, jika *timestamp* pada *invoice* tidak memenuhi kondisi tersebut, maka sistem akan mengambil harga *non earlybird* yang ditunjukkan pada baris 22-23. Kemudian pada baris 25 menunjukkan fungsi *view* akan menampilkan halaman *invoice\_view* yang tampilannya terlihat pada gambar 5.77 di atas.

### 5.10.3. Implementasi Fitur Pembuatan *Payment Receipt*

Implementasi fitur pembuatan *payment receipt* ini bertujuan untuk memenuhi kebutuhan penyelenggara ISICO dalam hal pembuatan bukti penerimaan pembayaran.

Receipt of Invoice 201703298Y682.pdf

1 / 1

**ISICO** Invoice 201703298Y682  
ITS

**[Invoice Paid!]**  
BRI to Marina Sudler  
Pansingling

Paper Registration				
Paper ID	Paper Title	Presenter	Type	Price
1	Early stage in computing with multimedia for learning	Marina Sudler	Early paper	2,000,000
2	Learning Training for Future Researcher Youtube Using Multimedia Evaluation	Marina Sudler	Additional paper	2,000,000

Participant Registration				
Name	Institution	Price		
Marina Sudler	ITS	2,000,000		

Hotel Booking Information					
Name	Room Type	Number of Rooms	Check In	Check Out	Price per room
Marina Sudler	Double	1	2017-03-29	2017-03-30	2,000,000

Proceeding Order

**Gambar 5.80** Tampilan fitur *payment receipt*

Gambar 5.80 merupakan tampilan fitur pembuatan *payment receipt* yang sudah siap untuk diunduh. Potongan kode untuk

*payment receipt* hampir sama dengan kode untuk pembuatan *invoice*. Potongan kode pada bagian *controller* adalah yang terlihat pada Gambar 5.81.

```

1.  function receipt()
2.  $this->load->library('/tcpdf/tcpdf');
3.  $judulpage = "View invoice";
4.  $data['judulpage'] = $judulpage;
5.  $this->load->model('model_invoice');
6.  $user_id = $this->uri->segment(3);
7.  $data['invoice'] = $this->model_invoice->
   >get_invoice($user_id)->result();
8.  $data['invoice_proceeding'] = $this->
   >model_invoice->get_proceeding($user_id)-
   >result();
9.  if      (strtotime($data['query_date'])[0]-
   >timestamp) <      strtotime("2017-03-08
   14:27:03")){
10. $data['query_paper'] = $this->
   >model_invoice->get_paper($user_id)-
   >result();
11. $data['query_price'] = $this->
   >model_invoice->get_price_eb($user_id)-
   >result();
12. }
13. else {
14. $data['query_price'] = $this->
   >model_invoice->get_price($user_id)-
   >result();
15. $data['query_paper'] = $this->
   >model_invoice->get_paper_ne($user_id)-
   >result();

```

```
16.  }  
17.  $this->load->view('makereceipt',$data);  
18.  }
```

**Gambar 5.81** Potongan kode pada *controllers payment receipt*

Pada potongan kode pada Gambar 5.81, sistem akan menyimpan semua data pada variabel *array*. Baris kedua berfungsi untuk memanggil *library* pembuatan Pdf. Baris ketiga mendefinisikan judul halaman yang kemudian disimpan pada *array* di baris keempat. Baris kelima menunjukkan bahwa sistem akan mengambil model *invoice* yang akan digunakan pada fungsi *payment receipt* ini.

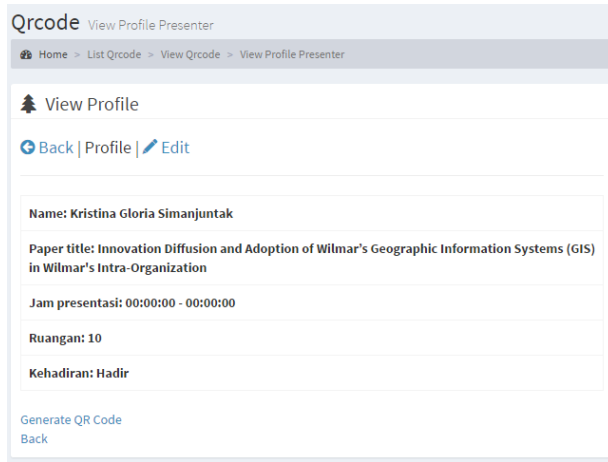
Kemudian, pada baris ke-6 menunjukkan bahwa sistem menyimpan segmen ketiga pada URL dalam sebuah variabel *user\_id*. Baris ke-7 sampai baris ke-8 menunjukkan bahwa sistem mengambil *query* yang tersimpan pada *model invoice*.

Lalu dari baris 9-15 merupakan fungsi penentuan *earlybird* atau tidak. Pada baris ke-9, sistem akan melakukan fungsi jika-maka. Hal ini ditujukan untuk melakukan perhitungan *earlybird* dengan menggunakan *timestamp* yang tersimpan pada tabel *invoice* dalam basis data. Dari fungsi pada baris ke-9 tersebut menunjukkan bahwa jika *timestamp* pada *invoice* kurang dari “2017-03-08 14:27:03”, maka sistem akan mengambil harga *earlybird* yang ditunjukkan pada baris 10-11. Namun, jika *timestamp* pada *invoice* tidak memenuhi kondisi tersebut, maka sistem akan mengambil harga *non earlybird* yang ditunjukkan pada baris 14-15. Kemudian pada baris 27 menunjukkan fungsi *view* akan menampilkan halaman *make\_receipt* yang tampilannya terlihat pada gambar 5.80 di atas.



#### 5.10.4. Implementasi Fitur Pembuatan *Qrcode*

Pembuatan fitur *qrcode* bertujuan untuk memenuhi kebutuhan penyelenggara ISICO dalam menggunakan *qrcode* sebagai tiket masuk saat konferensi berlangsung. *Qrcode* ini diberikan kepada masing-masing peserta, baik *author (presenter)* atau partisipan.



**Gambar 5.82 Tampilan profil *presenter* pada *qrcode***

Gambar 5.82 di atas merupakan tampilan *profile* dari *presenter* yang nantinya akan disimpan pada *qrcode*. Untuk melakukan pembuatan *qrcode*, pengguna harus mengklik *link generate qrcode*. Gambar 5.83 berikut adalah potongan kode yang digunakan ketika *link generate qrcode* diklik yang tersimpan pada *views qrcode\_author*.

```
1. <?=anchor('generateqrcode/index/'. $query_p
   rofila[0]->author_id,'Generate QR Code
```

**Gambar 5.83 Potongan *syntax* untuk membuat aksi *link***

Pada baris pertama menggunakan *syntax anchor* pada *framework codeigniter* untuk mengalihkan ke fungsi

*generateqrcode/index*. Lalu Gambar 5.82 di bawah ini merupakan *controllers* yang memproses terbentuknya *qrcode*.

```

1.  public function index (){
2.  $this->load->library('/qrcode-
    lib/ciqrcline');
3.  $this->load->helper('download');
4.  $user_id = $this->uri->segment(3);
5.  $params['data'] =
    'http://localhost/isico/index.php/qrcode/v
    iew_author_profile/' . $user_id;
6.  $params['level'] = 'H';
7.  $params['size'] = 3;
8.  $params['savename'] = FCPATH . 'tes.png';
9.  $this->ciqrcline->generate($params);
10. $data =
    file_get_contents(base_url() . 'tes.png');
11. $nama = 'qrcode_' . $user_id . '.png';
12. force_download($nama, $data);
13. }
```

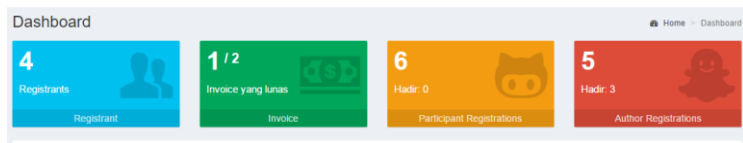
**Gambar 5.84** Potongan kode pada *controllers* pembuatan *qrcode*

Perlu diketahui bahwa pembuatan *qrcode* ini menggunakan sebuah *library* sehingga fungsi di sini akan memanggil *library* tersebut. Pada Gambar 5.84 di atas, baris kedua berfungsi untuk memanggil *library*. Baris ketiga menunjukkan bahwa sistem menggunakan *helper* dari *codeigniter*, yaitu *download* yang dalam hal ini berfungsi untuk melakukan pengunduhan *qrcode*. Baris keempat menunjukkan bahwa segmen ketiga dari URL disimpan pada variabel *user\_id*. Baris 6-8 menunjukkan bahwa sistem menyimpan beberapa parameter yang digunakan pada *qrcode*, seperti *level*, *size*, dan *savename*. Baris ke-9 merupakan fungsi utama, yaitu untuk melakukan *generating* terhadap parameter-parameter yang telah didefinisikan sebelumnya. Baris ke-10 menunjukkan URL

yang disimpan sebagai alamat yang dituju saat *qrcode* ini dipindai. Alamat ini akan mengarahkan pada *profile author* tadi. Kemudian baris 11 menunjukkan nama penyimpanan *file qrcode*, lalu pada baris ke-12 menunjukkan fungsi pengunduhan *qrcode*.

#### 5.10.5. Implementasi Fitur Informasi *Dashboard*

Pembuatan fitur informasi *dashboard* bertujuan untuk memenuhi kebutuhan penyelenggara ISICO untuk menampilkan informasi jumlah *author* yang terdaftar, jumlah *participant* yang terdaftar, jumlah peserta yang melunasi pembayaran, dan jumlah peserta yang telah hadir dalam konferensi.



**Gambar 5.85 Tampilan *dashboard* informasi**

Gambar 5.85 di atas merupakan tampilan *dashboard* yang dibangun. Bagian yang berwarna biru menampilkan informasi *registrants*/peserta yang terdaftar. Bagian yang berwarna hijau menampilkan jumlah pembayaran yang telah dilunasi per jumlah pembayaran seluruhnya yang ada. Bagian berwarna oranye menampilkan informasi jumlah partisipan yang terdaftar serta jumlah partisipan yang telah menghadiri konferensi. Bagian yang berwarna merah menampilkan informasi jumlah *author* yang terdaftar dan yang telah menghadiri konferensi. Potongan kode untuk *dashboard* pada bagian *controller* terlihat pada Gambar 5.86 di bawah ini.

Pada baris ke 2-4, fungsi ini akan mengambil model *invoice*, *model registrant*, dan *model dashboard*. Baris ke 5-6 menunjukkan bahwa fungsi ini membuat judul halaman pada baris ke-5 yang kemudian disimpan pada variabel di baris ke-

6. Baris 7-8 menunjukkan bahwa sistem mengambil *query* yang tersimpan pada *model invoice*, *registrant*, serta *dashboard*. Kemudian pada baris ke-9 fungsi ini menampilkan *view dashboard* yang diikuti dengan variabel *array* bernama *data*.

```
1. public function index (){
2.     $this->load->model('model_invoice');
3.     $this->load->model('model_registrant');
4.     $this->load->model('model_dashboard');
5.     $judulpage = "Daftar invoice";
6.     $data['judulpage'] = $judulpage;
7.     $data['invoice'] = $this->model_invoice-
        >list_invoice()->result();
8.     $data['registrant'] = $this-
        >model_registrant->list_registrant()-
        >result();
9.     $this->load->view('Dashboard',$data);
10. }
```

**Gambar 5.86** Potongan kode pada *controllers* fitur *dashboard*

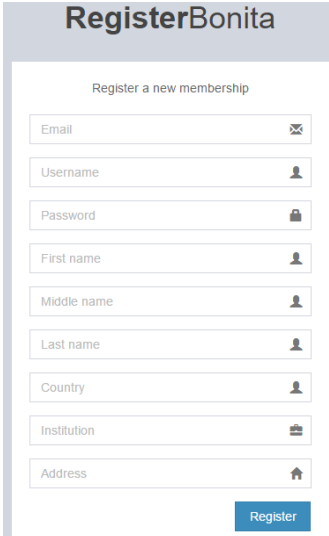
## BAB VI

### HASIL DAN PEMBAHASAN

Setelah melakukan implementasi model *to-be* ke dalam aplikasi Bonita BPM, serta telah dilakukan beberapa konfigurasi yang mendukung berjalannya proses agar sesuai harapan, berikut ini adalah hasil dari program yang telah dijalankan melalui Bonita Portal.

#### 6.1. Mendaftarkan Akun Bonita BPM

Sebelum melakukan pendaftaran ISICO, peserta harus mendaftarkan diri untuk mendapatkan akun Bonita BPM. Gambar 6.1 merupakan tampilan untuk mendaftarkan diri.



The image shows a web form titled "RegisterBonita" with the subtitle "Register a new membership". The form contains the following fields: Email (with an envelope icon), Username (with a person icon), Password (with a lock icon), First name (with a person icon), Middle name (with a person icon), Last name (with a person icon), Country (with a person icon), Institution (with a building icon), and Address (with a house icon). A blue "Register" button is located at the bottom right of the form.

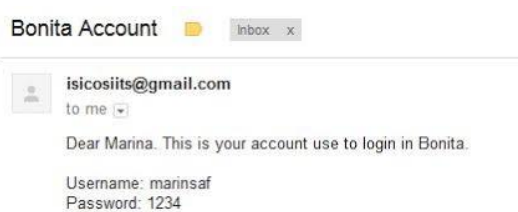
**Gambar 6.1 Mendaftar akun bonita**

Pendaftar harus mengisikan *email*, *username*, *password*, *first name*, *middle name*, *last name*, *country*, *institution*, dan *address*. Setelah pendaftar mendaftar, sistem akan menyimpan data pada tabel *registrant* di *database*. Gambar 6.2 merupakan hasil penyimpanan data pendaftaran akun.

user_id	user_email	username	user_pass	user_fname	user_midname	user_lname	user_country	user_institution	user_address
1	delinare.its@gmail.com	delinare	123	Delina	Rahayu	Effendi	Indonesia	ITS	Dukuh setro
2	marinavisafitri@gmail.com	marinasaf	1234	Marina		Salitri	Indonesia	ITS	Pandegling
3	shaniaoliviazayin@gmail.com	oliviazayin	123	Shania	Olivia	Zayin	Indonesia	ITS Sby	Perumdos
4	chandrasw.cs@gmail.com	chandrasw	123	Chandra	Surya	Wicaksono	Indonesia	ITS	Gebang
5	stezarp@gmail.com	stezarp	123	Stezar		Priansya	Indonesia	ITS	Karanggayam

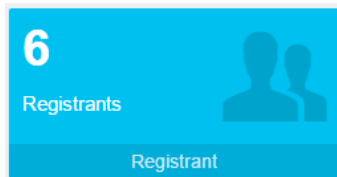
**Gambar 6.2 Hasil penyimpanan data pendaftaran akun**

Setelah itu, bagian keuangan akan mendaftarkan pendaftar baru tersebut untuk dapat memiliki akun pada Bonita. Lampiran F Gambar F.1 merupakan tampilan formulir yang harus diisi. Ketika bagian keuangan melakukan *submit*, sistem akan mengirimkan pesan pemberitahuan kepada pendaftar bahwa pendaftar terkait telah memiliki akun pada Bonita BPM dan dapat digunakan untuk melakukan proses pendaftaran ISICO. Gambar 6.3 merupakan hasil pemberitahuan yang diterima oleh pendaftar.



**Gambar 6.3 Hasil pemberitahuan email akun telah terdaftar**

Pada *dashboard*, juga melakukan pembaruan terhadap jumlah pendaftar. Gambar 6.4 merupakan tampilan hasil pembaruan jumlah pendaftar (*registrant*) pada *dashboard*.



**Gambar 6.4 Hasil pembaruan dashboard**

## 6.2. Memilih Mendaftarkan *Paper* atau Tidak

Proses memilih ini merupakan aktivitas awal dari pendaftaran ISICO. Seperti yang terlihat pada Lampiran F Gambar F.2, pendaftar harus memilih salah satu pilihan, *yes* atau *no*. Jika pendaftar memilih *yes*, pendaftar akan melanjutkan aktivitas menambahkan *paper*. Jika pendaftar memilih *no*, maka pendaftar akan melanjutkan ke aktivitas memilih menambah partisipan.

## 6.3. Menambahkan *Paper*

Setelah pendaftar memilih *yes* saat berada di aktivitas memilih mendaftar *paper* atau tidak, pendaftar di sini akan menambahkan *paper* dengan memasukkan ID *paper* yang terlihat pada Lampiran F Gambar F.3.

Saat ID *paper* disubmit, sistem akan memperbarui data *paper* pada tabel *author* dalam *database*. Dalam pembaruan ini, *database* memperbarui dengan menambahkan *user\_id* yang mendaftar *paper* tersebut. Pada contoh simulasi yang dilakukan penulis ini, *user* ID yang digunakan adalah 2 dan *paper* ID adalah 4, sehingga dalam *database* akan memperbarui semua *user* ID dengan 2 yang memiliki *paper* dengan ID 4 seperti yang terlihat pada Gambar 6.5.

<i>user_id</i>	<i>author_name</i>	<i>author_membership</i>	<i>paper_id</i>
2	Rianto		4
2	Ridi Ferdiana		4
2	Marius Limpo	Student	4

Gambar 6.5 Hasil pembaruan *database* terkait *paper*

## 6.4. Memilih *presenter*

Setelah memasukkan ID *paper*, pendaftar akan memasuki aktivitas memilih *presenter*. Dalam aktivitas ini terdapat beberapa *database connector* untuk mengambil jenis

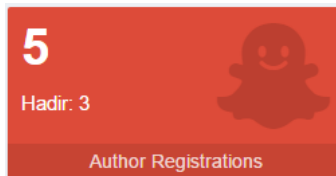
*membership*, nama *presenter* pada *paper* terkait, dan judul *paper*. Hasil pengambilan data tersebut dapat dilihat saat proses dijalankan, seperti pada Lampiran F Gambar F.4.

Setelah *presenter* terpilih dan peserta melakukan *submit*, sistem akan memperbarui data *paper* pada tabel *author* dalam *database*. Pada contoh simulasi ini, yang terpilih menjadi *presenter* untuk *paper* ID 4 adalah Marius Limpo, sehingga hasil pembaruan pada *database* terlihat seperti Gambar 6.6. Karena *paper* ini merupakan *main paper*, maka sistem hanya memperbarui *flag\_presenter* dengan angka 1, dimana menandakan bahwa Marius merupakan *presenter* pada *main paper* dengan ID 4.

<i>user_id</i>	<i>author_name</i>	<i>author_membership</i>	<i>paper_id</i>	<i>flag_presenter</i>	<i>flag_additional</i>
2	Rianto		4	0	0
2	Ridi Ferdiana		4	0	0
2	Marius Limpo	Student	4	1	0

**Gambar 6.6** Hasil pembaruan *database* terkait *presenter*

Pada *dashboard*, juga melakukan pembaruan terhadap jumlah *author* yang terdaftar. Gambar 6.7 merupakan tampilan hasil pembaruan jumlah *author* pada *dashboard*.



**Gambar 6.7** Hasil pembaruan *author registrations* pada *dashboard*

## 6.5. Menambahkan *Paper* Tambahan

Saat pendaftar memilih untuk menambahkan *paper* tambahan, pendaftar akan masuk pada aktivitas menambahkan *paper* tambahan ini. Dalam aktivitas ini terdapat *database connector* untuk mengambil jumlah sisa *paper* yang dimiliki oleh *presenter* yang terpilih pada aktivitas memilih *presenter*, dan



mengambil judul *paper* yang tersisa yang dimiliki oleh *presenter* tersebut. Hal ini dapat dilihat saat proses dijalankan, seperti yang terlihat pada Lampiran F Gambar F.5.

Setelah pendaftar melakukan *submit* pada aktivitas ini, sistem akan memperbarui data *paper* pada tabel *author* dalam *database*. Pada contoh simulasi ini, karena *paper* yang tersisa dengan *presenter* Marius Limpo, maka *database* memperbarui data tersebut dengan menambahkan *user ID*, *flag\_presenter*, dan *flag\_additional*, karena *paper* ini merupakan *paper* tambahan seperti yang terlihat pada Gambar 6.8.

<i>user_id</i>	<i>author_name</i>	<i>author_membership</i>	<i>paper_id</i>	<i>flag_presenter</i>	<i>flag_additional</i>
2	Rianto		4	0	0
2	Ridi Ferdiana		4	0	0
2	Marius Limpo	Student	5	1	1
2	Marius Limpo	Student	4	1	0

**Gambar 6.8** Hasil pembaruan *database* terkait penambahan *additional paper*

## 6.6. Memilih Menambahkan Partisipan atau Tidak

Proses memilih ini merupakan aktivitas yang terjadi setelah pendaftar selesai mendaftarkan *paper* atau pendaftar memilih *no* saat berada di aktivitas memilih mendaftarkan *paper* atau tidak. Seperti yang terlihat pada Lampiran F Gambar F.6, pendaftar harus memilih salah satu pilihan, *yes* atau *no*. Jika pendaftar memilih *yes*, pendaftar akan melanjutkan aktivitas menambahkan partisipan. Jika pendaftar memilih *no*, maka pendaftar akan melanjutkan ke aktivitas memilih menambahkan kebutuhan lain atau tidak.

## 6.7. Menambahkan Partisipan

Ketika pendaftar memilih *yes* saat memilih menambahkan partisipan atau tidak, pendaftar akan melakukan pendaftaran partisipan atau *audience* yang hadir dalam ISICO. Pada aktivitas ini, pendaftar akan mengisi *first name*, *middle name*,

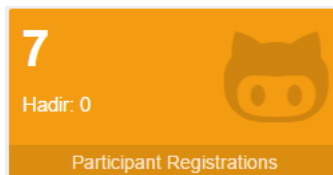
*last name*, dan institusi dari partisipan yang didaftarkan. Formulir untuk menambahkan partisipan dapat dilihat pada Lampiran F Gambar F.7.

Setelah pendaftar melakukan *submit* pada aktivitas menambahkan partisipan ini, sistem akan melakukan penyimpanan data partisipan dalam *database* pada tabel *add participant*. Pada contoh simulasi yang dilakukan penulis ini, penulis menambahkan sebanyak dua partisipan untuk mengikuti ISICO, sehingga hasil penyimpanan data dalam *database* seperti yang terlihat pada Gambar 6.9.

user_id	participant_fname	participant_midname	participant_lname	participant_institusi
2	Stezar		Priansya	ITS
2	Pramita	Lucianna	P. S.	ITS

**Gambar 6.9 Hasil pembaruan *database* terkait pendaftaran partisipan**

Pada *dashboard*, juga melakukan pembaruan terhadap jumlah partisipan yang terdaftar. Gambar 6.10 merupakan tampilan hasil pembaruan jumlah partisipan pada *dashboard*.



**Gambar 6.10 Hasil pembaruan jumlah partisipan pada *dashboard***

## **6.8. Memilih Menambahkan Kebutuhan Lain atau Tidak**

Proses memilih ini merupakan aktivitas yang terjadi setelah pendaftar selesai mendaftarkan partisipan atau peserta memilih *no* saat berada di aktivitas memilih mendaftarkan partisipan atau tidak. Seperti yang terlihat pada Gambar F.8, pendaftar harus memilih salah satu pilihan, *yes* atau *no*. Jika pendaftar memilih *yes*, pendaftar akan melanjutkan aktivitas

menambahkan kebutuhan lain. Di sini, pendaftar dapat memesan prosiding atau mereservasi hotel, atau keduanya. Jika pendaftar memilih *no*, maka pendaftar telah selesai melakukan pendaftaran.

### 6.9. Memesan Prosiding

Saat pendaftar memilih menambahkan kebutuhan lain, pendaftar pertama kali akan memasuki aktivitas memesan prosiding. Jika pendaftar ingin memesan, pendaftar harus memilih berapa banyak prosiding yang ingin dipesan sesuai dengan pilihan yang ada. Namun bila pendaftar hanya ingin mereservasi prosiding, pendaftar tidak perlu memilih dan membiarkan *field* prosiding seperti semula. Lampiran F Gambar F.9 merupakan tampilan pemesanan prosiding.

Ketika pendaftar memesan prosiding, sistem akan menyimpan pemesanan prosiding ke dalam *database* pada tabel kebutuhan lain. Pada contoh simulasi yang dilakukan oleh penulis, penulis tidak memasukan pemesanan prosiding, sehingga sistem menyimpan jumlah 0 pada atribut *proceeding\_qty*. Gambar 6.11 merupakan hasil pembaruan *database* terkait pemesanan prosiding.

user_id	proceeding_qty
1	1
NULL	NULL
2	0

Gambar 6.11 Hasil pembaruan *database* terkait pemesanan prosiding

### 6.10. Mereservasi Hotel

Setelah melakukan pemesanan prosiding, pendaftar akan memasuki aktivitas layanan reservasi hotel ketika pendaftar mencentang pernyataan *hotel reservation* di halaman

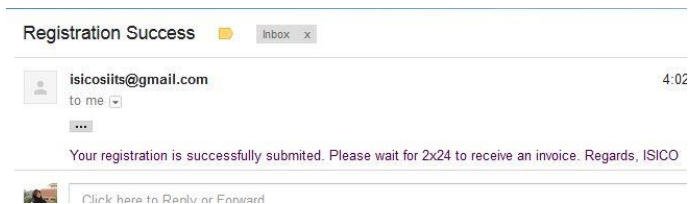
pemesanan prosiding. Seperti yang terlihat pada Lampiran F Gambar F.10, pendaftar harus mengisi *passport ID*, *full name*, *gender*, *country*, *hotel*, jumlah ruangan yang dipesan, jumlah tamu yang menginap, tanggal *check in*, dan tanggal *check out*.

Pada aktivitas mereservasi hotel ini, sistem akan melakukan penyimpanan data reservasi pada tabel kebutuhan lain di *database*. Hasil penyimpanan dapat dilihat pada Gambar 6.12.

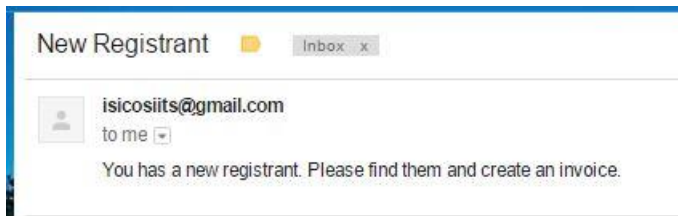
user_id	proceeding_qty	passport_id	full_name	gender	country	hotel_name	hotel_room	hotel_people	hotel_check_in	hotel_check_out
2	0	2	Marina Safitri	Female	Indonesia	Pool View	1	2	2017-04-12	2017-04-13

**Gambar 6.12 Hasil pembaruan *database* terkait data reservasi hotel**

Setelah menyelesaikan reservasi hotel, atau bila pendaftar tidak melakukan reservasi (hanya memesan prosiding), sistem akan mengirimkan pesan kepada pendaftar bahwa pendaftarannya telah selesai. Gambar 6.13 merupakan tampilan pemberitahuan *email* kepada pendaftar. Selain memberikan pemberitahuan kepada pendaftar, sistem juga memberi pemberitahuan kepada bagian keuangan agar bagian keuangan mengetahui bahwa ada pendaftar baru dan dapat segera memproses pendaftaran pendaftar tersebut. Gambar 6.14 merupakan tampilan pemberitahuan *email* kepada bagian keuangan.



**Gambar 6.13 Hasil pemberitahuan *email* kepada bagian keuangan**



**Gambar 6.14** Hasil pemberitahuan *email* kepada peserta

Selain mengirim pesan pemberitahuan, sistem juga melakukan pembaruan *invoice* dalam *database* pada tabel *invoice*. Gambar 6.15 adalah hasil pembaruan terhadap tabel *invoice*.

timestamp	id	invoice_id	user_id	lunas	note
2017-03-29 14:08:37	1	201703299YVKN1	1	1	NULL
2017-04-10 07:44:22	9		2	0	

**Gambar 6.15** Hasil pembaruan *database* terkait adanya *invoice* baru

## 6.11. Membuat *Invoice*

Aktivitas pembuatan *invoice* ini dilakukan di sistem eksternal. Aktivitas ini dilakukan setelah bagian keuangan menerima pemberitahuan bahwa terdapat pendaftar baru yang mendaftar ISICO sehingga harus dibuatkan *invoice*. Langkah untuk membuat *invoice* adalah masuk ke halaman menu *invoice* pada *dahsboard*. Gambar 6.16 merupakan hasil saat masuk pada menu *invoice*.

ID Invoice	ID User	Status	Action
201703299YVKN1	1	Lunas	<a href="#">View Invoice</a>
ID Tidak Tersedia	2	Belum Lunas	<a href="#">Create Invoice</a>

**Gambar 6.16** Daftar yang harus dibuatkan *invoice*

Untuk membuat *invoice*, klik *action link create invoice* pada ID 2, yaitu ID yang digunakan pada contoh simulasi ini. Saat

*link* tersebut diklik, sistem akan melakukan pembuatan *ID invoice* beserta *invoicenya*. *ID invoice* meliputi perpaduan tanggal, huruf dan angka acak, serta *user ID*. Gambar 6.17 adalah hasil pembuatan *invoice* beserta *ID invoice*.

The screenshot shows a web application titled 'List Invoice' with a 'View Invoice' button. Below the button are links for 'Back', 'View invoice', and 'Save as Pdf'. The invoice is addressed to 'Marina Safitri' and is in the 'Pandegiling' status. The invoice number is '201704102QITX2'. It lists two items under 'Paper Registration':

Paper ID	Paper Title	Presenter	Type	Price
4	Usability Testing on Flight Searching Website Using Heuristic Evaluation	Marius Limpo	Main paper	4,000,000
5	Web Tags Formatting with Multilevel Numbering	Marius Limpo	Additional paper	2,000,000

Below the items is a 'Participant Registration' table:

Name	Institution	Price

**Gambar 6.17 Hasil pembuatan *invoice***

Setelah itu, bagian keuangan harus mengklik *save as Pdf* untuk membuat *copy invoice* tersebut dalam bentuk Pdf. Gambar 6.18 merupakan hasil ketika *save as Pdf* diklik. Di sini, *file invoice* dapat diunduh.

The screenshot shows a PDF preview of the invoice. It includes the 'ISICO ITS' logo and the same data as the web interface. It also includes a 'Total Booking Information' section:

Name	Room Type	Number of Room	Check In	Check Out	Price per room
Marina Safitri	First view	5	2017-04-12	2017-04-13	\$50,000

There is also a 'Processing Order' section:

User ID	Processing Quantity	Price
	5	\$25,000

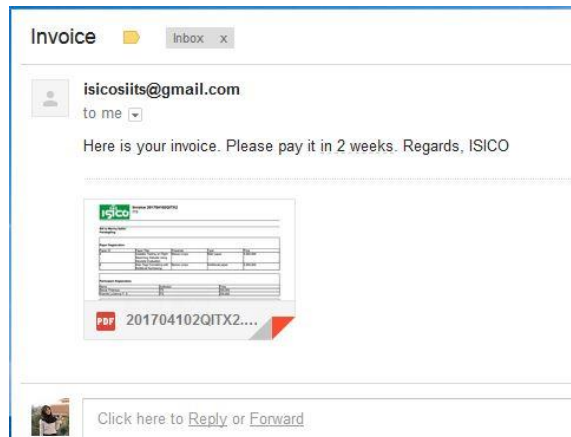
Finally, a 'Summary Registration' table:

Participant price	Processing	Total	Total Price
\$200,000	\$25,000	\$225,000	\$225,000

**Gambar 6.18 Preview *invoice* sebelum diunduh**

## 6.12. Mengirimkan *Invoice*

Setelah *file invoice* selesai diunduh, selanjutnya bagian keuangan melanjutkan aktivitas pada Bonita BPM, yaitu mengirimkan *invoice* kepada pendaftar. Lampiran F Gambar F.11 merupakan tampilan formulir untuk mengirimkan *invoice*. Bagian keuangan harus mengisikan *invoice* ID dan mengunggah *file invoice* pada sistem tersebut. Setelah itu, bagian keuangan mengirimkannya kepada pendaftar. Setelah bagian keuangan mengirim, sistem akan memberikan pesan pemberitahuan kepada pendaftar. Gambar 6.19 merupakan tampilan pemberitahuan yang diterima oleh pendaftar.



**Gambar 6.19** Hasil pengiriman *invoice* melalui *email*

## 6.13. Mengunggah Bukti Pembayaran

Setelah *email* diterima, pendaftar harus mengunggah bukti pembayaran. Di sini, bukti harus berupa *file* berformat Pdf. Gambar 6.20 merupakan tampilan bila *file* tidak berformat Pdf, sehingga sistem mengeluarkan pesan validasi. Lampiran F Gambar F.12 merupakan tampilan formulir mengunggah bukti pembayaran. Pada formulir tersebut, peserta akan mengetahui

nomor ID *invoice* serta mendapatkan lagi *file invoice* selain yang didapatkan pada *email*.

Setelah pendaftar melakukan *submit*, sistem akan mengirim pesan kepada bagian keuangan bahwa terdapat pendaftar yang telah menggunggah bukti pembayaran. Gambar 6.21 merupakan tampilan pemberitahuan yang diterima oleh bagian keuangan.

### Mengupload Bukti Pembayaran

Invoice id

201704102QITX2

Invoice

☐ URL ☒ File

[201704102QITX2.pdf](#)

[modify](#) [remove](#)

Payment Receipts \*

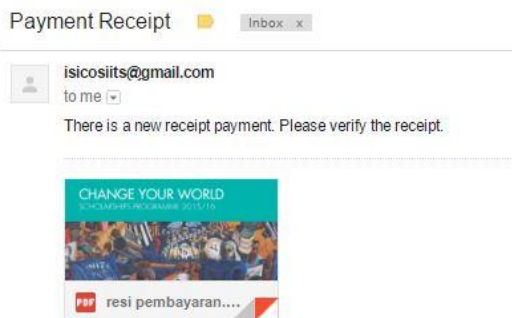
[qrcode\\_1 \(3\).png](#)

[modify](#) [remove](#)

File format must be \*.pdf

SUBMIT

**Gambar 6.20 Tampilan *validator* pada pengunggahan bukti pembayaran**

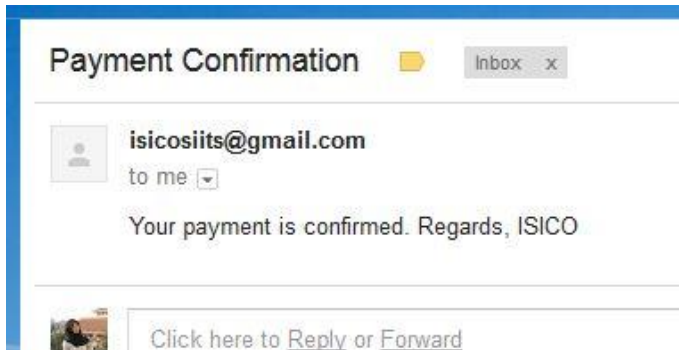


**Gambar 6.21 Hasil pengiriman bukti pembayaran**



#### 6.14. Meverifikasi Bukti Pembayaran

Setelah bagian keuangan menerima *email* bahwa terdapat pendaftar yang telah mengunggah bukti pembayaran, bagian keuangan akan melakukan verifikasi terhadap bukti pembayaran tersebut. Lampiran F Gambar F.13 merupakan tampilan untuk verifikasi bukti pembayaran. Saat bagian keuangan telah mengklik tombol *confirm*, sistem akan mengirimkan pesan kepada pendaftar bahwa pembayarannya telah dikonfirmasi. Gambar 6.22 merupakan tampilan pemberitahuan yang diterima oleh pendaftar.



**Gambar 6.22 Hasil pemberitahuan terkait verifikasi pembayaran**

Selain itu, sistem juga akan melakukan pembaruan pada tabel *invoice*, *author*, dan *add participant* pada *database*. Gambar 6.23 merupakan hasil pembaruan pada tabel *invoice*, dimana *flag\_kelunasan* menjadi 1 karena pembayaran telah dilunasi. Gambar 6.24 merupakan hasil pembaruan pada tabel *author*, dimana *flag\_kelunasan* pada *user* ID 2 menjadi 1 karena *paper* pada tersebut telah dilunasi. Gambar 6.25 merupakan hasil pembauran pada tabel *add participant*, dimana *flag\_kelunasan* menjadi 1 untuk *user* ID 2.

Pada *dashboard*, juga melakukan pembaruan terhadap jumlah *invoice* yang telah lunas. Gambar 6.26 merupakan tampilan hasil pembaruan pelunasan *invoice* pada *dashboard*.

timestamp	id	invoice_id	user_id	lunas	note
2017-03-29 14:08:37	1	201703299YVKN1	1	1	NULL
2017-04-10 07:44:22	9	201704102QITX2	2	1	lunas

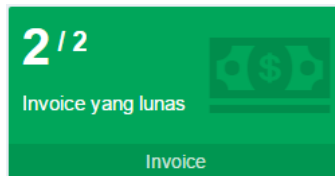
**Gambar 6.23** Hasil pembaruan *database* pada tabel *invoice*

user_id	author_name	author_membership	paper_id	flag_presenter	flag_additional	flag_kelunasan
2	Rianto		4	0	0	1
2	Ridi Ferdiana		4	0	0	1
2	Marius Limpo	Student	5	1	1	1
2	Marius Limpo	Student	4	1	0	1

**Gambar 6.24** Hasil pembaruan *database* pada tabel *author*

user_id	participant_fname	participant_midname	participant_lname	participant_institusi	flag_kelunasan
2	Stezar		Priansya	ITS	1
2	Pramita	Lucianna	P. S.	ITS	1

**Gambar 6.25** Hasil pembaruan *database* pada tabel partisipan



**Gambar 6.26** Hasil pembaruan *dashboard* terkait pelunasan *invoice*

## 6.15. Membuat *Qrcode* dan Bukti Penerimaan Pembayaran

Aktivitas pembuatan *qrcode* dan bukti penerimaan pembayaran ini dilakukan di sistem eksternal. Aktivitas ini dilakukan setelah bagian keuangan melakukan konfirmasi terhadap bukti pembayaran dari pendaftar. Pertama-tama, langkah untuk membuat *qrcode* adalah masuk ke halaman menu *qrcode* pada *dahsboard*. Gambar 6.27 merupakan hasil saat masuk pada menu *qrcode*.

Qrcode Generator			
List Qrcode			
ID Invoice	ID User	Status	Action
201703299YVKN1	1	Lunas	<a href="#">View Invoice</a>
201704102QITX2	2	Lunas	<a href="#">View Invoice</a>

**Gambar 6.27 Daftar yang harus dibuatkan *qrcode***

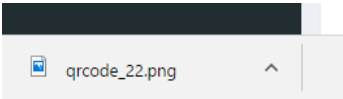
Untuk membuat *qrcode*, klik *action link view invoice* pada ID 2, yaitu ID yang digunakan pada contoh simulasi ini. Saat *link* tersebut diklik, sistem akan mengarahkan pada halaman *view invoice* namun tampilannya sedikit berbeda dengan *invoice*. Gambar 6.28 merupakan tampilan *view invoice* saat akan membuat *qrcode*.

List Qrcode View Qrcode				
View Qrcode				
<a href="#">Back</a>   <a href="#">Daftar QrCode</a>   <a href="#">Download receipt</a>				
Author				
Name	Paper Title	Kehadiran	Action	Qrcode
Marius Limpo	Web Tags Formatting with Multilevel Numbering	Belum Hadir	<a href="#">View profile</a>	<a href="#">Generate QR Code</a>
Marius Limpo	Usability Testing on Flight Searching Website Using Heuristic Evaluation	Belum Hadir	<a href="#">View profile</a>	<a href="#">Generate QR Code</a>
Participant Registration				
Name	Institution	Kehadiran	Action	Qrcode
Stezar Priansya	ITS	Belum Hadir	<a href="#">View profile</a>	<a href="#">Generate QR Code</a>
Pramita Lucianna P. S.	ITS	Belum Hadir	<a href="#">View profile</a>	<a href="#">Generate QR Code</a>

**Gambar 6.28 Tampilan saat akan membuat *qrcode***

Untuk membuat *qrcode*, maka perlu melakukan klik pada *generate qrcode* pada masing-masing nama. Hasilnya adalah berupa *file qrcode* berformat Png. Gambar 6.29 merupakan hasil pengunduhan *qrcode* dan Gambar 6.30 merupakan hasil *qrcode* untuk nama yang dilakukan *generate qrcode* tadi.

Untuk membuat *receipt* atau bukti penerimaan, maka pada *view invoice* Gambar 6.28, klik pada *link download receipt*. Gambar 6.31 merupakan hasil tampilan *receipt* setelah *link* tersebut diklik.



Gambar 6.29 Hasil pengunduhan *qrcode*



Gambar 6.30 Hasil *qrcode*



Gambar 6.31 *Preview receipt* sebelum diunduh

Pada gambar tersebut, tertulis tulisan *invoice paid*, yang berarti *invoice* dengan nomor *invoice* tersebut telah terbayar lunas. *File* tersebut dapat diunduh untuk kemudian dikirimkan.

## 6.16. Mengirim *Qrcode* dan Bukti Penerimaan Pembayaran

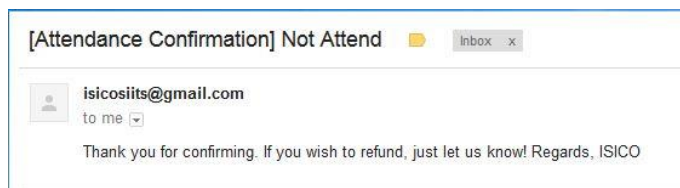
Setelah bagian keuangan telah mengunduh semua *file invoice* dan *qrcode*, bagian keuangan harus mengirimkannya kepada pendaftar. Lampiran F Gambar F.14 merupakan tampilan formulir untuk mengirimkan *invoice* dan *qrcode* kepada pendaftar. Saat semua *file* tersebut terkirim, pendaftar akan menerima pemberitahuan dari *email*. Gambar 6.32 merupakan tampilan pemberitahuan yang diterima oleh pendaftar.



**Gambar 6.32 Hasil pemberitahuan dan pengiriman *qrcode* dan bukti penerimaan pembayaran**

## 6.17. Konfirmasi Kehadiran

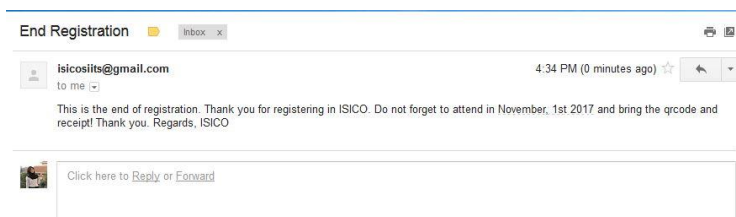
Aktivitas selanjutnya setelah peserta menerima *email* beserta *receipt* dan *qrcode* adalah peserta melakukan konfirmasi kehadiran. Lampiran F Gambar F.15 merupakan tampilan formulir untuk melakukan konfirmasi kehadiran. Jika pendaftar memilih *yes*, pendaftar akan melanjutkan aktivitas untuk konfirmasi kebenaran nama. Namun jika pendaftar memilih *no*, proses telah berakhir dan sistem akan mengirim pesan kepada pendaftar. Gambar 6.33 merupakan hasil pengiriman pesan kepada pendaftar.



**Gambar 6.33 Hasil pemberitahuan terkait ketidakhadiran peserta**

## 6.18. Konfirmasi Kebenaran Nama

Bila pendaftar memilih *yes* pada saat konfirmasi kehadiran, pendaftar akan lanjut pada aktivitas konfirmasi kebenaran nama ini. Lampiran F Gambar F.16 merupakan tampilan untuk konfirmasi kebenaran nama. Setelah pendaftar melakukan konfirmasi kebenaran nama, pendaftar akan menerima pesan pemberitahuan bahwa seluruh proses pendaftaran telah selesai. Gambar 6.34 merupakan tampilan pemberitahuan yang diterima oleh pendaftar.



**Gambar 6.34 Hasil pemberitahuan selesainya proses pendaftaran**

## 6.19. Mengisi Formulir Pembetulan Nama

Ketika peserta memilih ingin melakukan pembetulan nama, pendaftar akan melanjutkan pada aktivitas mengisi formulir pembetulan nama seperti pada Lampiran F Gambar F.17. Peserta harus mengisi ID nama yang ingin diperbarui namanya. ID ini dapat dilihat melalui *invoice/receipt* yang telah diterima. Selain itu, pendaftar juga harus mengisi nama pembetulan yang terdiri dari *first name*, *middle name*, dan *last name* serta mengisi tipe pendaftar untuk nama terkait,

*presenter* atau *participant*. Setelah pendaftar melakukan *submit*, sistem akan melakukan penyimpanan pada *database*. Gambar 6.35 adalah contoh hasil penyimpanan yang dilakukan.

name_id	first_name	middle_name	last_name	type
23	Pramita	Lucianna	Putri	Participant

**Gambar 6.35 Hasil penyimpanan data pembenaran nama**

*Halaman ini sengaja dikosongkan*



## **BAB VII**

### **KESIMPULAN DAN SARAN**

Dalam bab ini akan dijelaskan mengenai kesimpulan dan saran yang dihasilkan dari penelitian ini.

#### **7.1. Kesimpulan**

Setelah menyelesaikan tugas akhir terkait otomasi sistem pendaftaran ISICO, dapat ditarik kesimpulan pada proses sebagai berikut.

1. Melalui pemetaan permasalahan yang dihadapi penyelenggara ISICO dengan proses yang diinginkan terkait pendaftaran ISICO, ditemukan beberapa perubahan yang dilakukan, yaitu sebagai berikut.
  - a. Mengotomasi proses pendaftaran ISICO yang mulanya manual dan dikerjakan secara terpisah-pisah pada sistem pendaftaran ISICO dengan Bonita BPM.
  - b. Melakukan pembuatan *invoice*, *qr code*, *receipt* melalui sistem eksternal sehingga bagian keuangan tidak perlu melakukannya secara manual.
  - c. Adanya *qr code* yang dapat digunakan sebagai pengganti absensi manual untuk melakukan registrasi ulang saat konferensi berlangsung.
2. Implementasi menggunakan Bonita BPM dapat mengotomasi proses pendaftaran ISICO yang dibuktikan dengan aktivitas-aktivitas yang dapat berjalan dalam Bonita BPM sebanyak 16 aktivitas.

Adapun kesimpulan hasil yang dapat ditarik dalam tugas akhir ini adalah sebagai berikut.

1. Terdapat beberapa kebutuhan yang tidak dapat dibangun melalui Bonita BPM dikarenakan fitur tersebut tidak terfasilitasi pada *free version* Bonita BPM. Hal ini menyebabkan diperlukannya untuk membangun sistem eksternal.

2. Sistem yang dibangun telah memenuhi kebutuhan dari proses pendaftaran ISICO. Kebutuhan tersebut telah terfasilitasi dari sistem yang dibangun dengan Bonita maupun sistem eksternal.
3. Dengan adanya sistem pendaftaran ISICO ini, data pendaftaran peserta dapat saling terintegrasi dalam sebuah *database*.

## 7.2. Saran

Saran yang dapat diusulkan penulis untuk penelitian yang akan dilakukan selanjutnya adalah melakukan analisa kebutuhan studi kasus yang akan diotomasi menggunakan Bonita BPM dengan fitur-fitur yang tersedia pada Bonita BPM. Hal ini dikarenakan terdapat beberapa fitur yang tidak dapat memenuhi kebutuhan proses pada versi *open source* atau *free version*, sehingga penulis perlu untuk membuat sistem eksternal.

Adapun saran yang dapat dilakukan untuk pengembangan aplikasi pada tugas akhir ini adalah sebagai berikut.

1. Melakukan konfigurasi sistem ke dalam sebuah *server* agar memudahkan pengguna untuk mengakses.
2. Menambahkan fitur untuk mengunduh atau menampilkan informasi aktivitas yang sedang dijalani oleh pendaftar pada sistem eksternal untuk pemantauan aktivitas yang dilakukan pendaftar.
3. Melakukan perbaikan pembuatan *qr code* yang langsung digabung dengan *receipt*.
4. Melakukan konfigurasi agar data pada aktivitas yang sudah dilakukan pendaftar tersimpan dalam *database* sehingga pendaftar tidak perlu mengulang proses dari awal.

## DAFTAR PUSTAKA

- [1] Jurusan Sistem Informasi, Fakultas Teknologi Informasi, ITS, “ISICO 2013,” 15 February 2013. [Online]. Available: <http://is.its.ac.id/scholar/?p=134>. [Diakses 22 September 2016].
- [2] Mustakim, Guntoro dan dkk, “Perancangan Sistem Informasi Pendaftaran Seminar Mahasiswa Pascasarjana Institut Pertanian Bogor,” *Jurnal Rekayasa dan Manajemen Sistem Informasi*, vol. 1, no. 2, pp. 6-14, 2015.
- [3] Junaidi, A. Roji dan K. Munawar, “Konsep Otomatisasi Sistem Pembayaran SPP Online untuk Mengurangi Tingkat Keterlambatan,” dalam *Konferensi Nasional Sistem & Informatika (KNS&I)*, Bali, 2015.
- [4] J. Satriawan dan A. Wibisono, Pengembangan Sistem Reservasi Kendaraan dengan Menggunakan Pendekatan Manajemen Proses Bisnis (Studi Kasus: PT. Pertamina EP), Surabaya, 2016.
- [5] T. Singer, “Business Strategy,” 19 June 2013. [Online]. Available: <https://www.mastercardbiz.com/2013/06/19/5-reasons-you-should-attend-a-conference-in-your-field/>. [Diakses 10 October 2016].
- [6] S. E. Smith dan C. T. Rankin, “Conferences: Why to Attend and How to Benefit,” University of Texas, Austin.
- [7] Wikipedia, “Information Systems International Conference,” 2 February 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Information\\_Systems\\_International\\_Conference](https://en.wikipedia.org/wiki/Information_Systems_International_Conference). [Diakses 24 September 2016].
- [8] P. Tony D. Susanto, “Information Systems International

- Conference - ISICO,” 18 Nopember 2013. [Online]. Available:  
[https://web.facebook.com/ISICO.info/posts/585636624847311?\\_rdr](https://web.facebook.com/ISICO.info/posts/585636624847311?_rdr). [Diakses 25 September 2016].
- [9] S. A. White, “Introduction to BPMN,” IBM Corporation, United State.
- [10] Wikipedia, “Bonita BPM,” 1 June 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Bonita\\_BPM](https://en.wikipedia.org/wiki/Bonita_BPM). [Diakses 27 September 2016].
- [11] Bonitasoft, “Products,” 2001-2016. [Online]. Available: <http://www.bonitasoft.com/products>. [Diakses 27 September 2016].
- [12] M. Dumas, M. L. Rosa dan dkk, Fundamentals of ss Process Managment, London: Springer Heidelberg ork, 2013.
- [13] S. Nidhra dan J. Dondeti, “Black Box and White Box Testing Techniques - A Literature Review,” *International Journal of Embedded System and Applications*, vol. 2, no. 2, p. 3, 2012.
- [14] A. Meidan, J. A. Garcia-Garcia dan dkk, “A Survey on Business Processes Management Suites,” *Computer Standards & Interfaces*, 2016.
- [15] Indonesia Productivity And Quality Institute, October 2015. [Online]. Available: [https://www.proxsis.com/wp-content/uploads/2015/02/News\\_IPQI-oktober2\\_opt.pdf](https://www.proxsis.com/wp-content/uploads/2015/02/News_IPQI-oktober2_opt.pdf). [Diakses 26 September 2016].
- [16] Redstone Software, “Black-box vs. White-box Testing: Choosing the Right Approach to Deliver Quality

Applications,” 2008.

- [17] P. C. Jorgensen, Software Testing - A Craftman's Approach, Boca Raton: Taylor & Francis Group, LLC, 2014.
- [18] W. E. Lewis, Software Testing and Continuous Quality Improvement Third Edition, Boca Raton: Taylor & Francis Group, LLC, 2009.

*Halaman ini sengaja dikosongkan*

## Lampiran A. Hasil Wawancara Proses *as-is* ISICO

Tabel A.1 berisikan informasi wawancara sedangkan Tabel A.2 berisikan hasil wawancara dengan narasumber, yaitu penyelenggara ISICO.

**Tabel A.1 Informasi wawancara proses *as-is***

Informasi Wawancara	
<b>Narasumber</b>	Amna Shifia Nisafani
<b>Jabatan</b>	Bagian Registrasi ISICO 2017
<b>Pewawancara</b>	Delina Rahayu Effendi
<b>Jabatan</b>	Peneliti
<b>Tempat</b>	
<b>Tanggal</b>	27 September 2016
<b>Tujuan</b>	Menggali informasi mengenai kondisi proses pendaftaran saat ini pada ISICO.

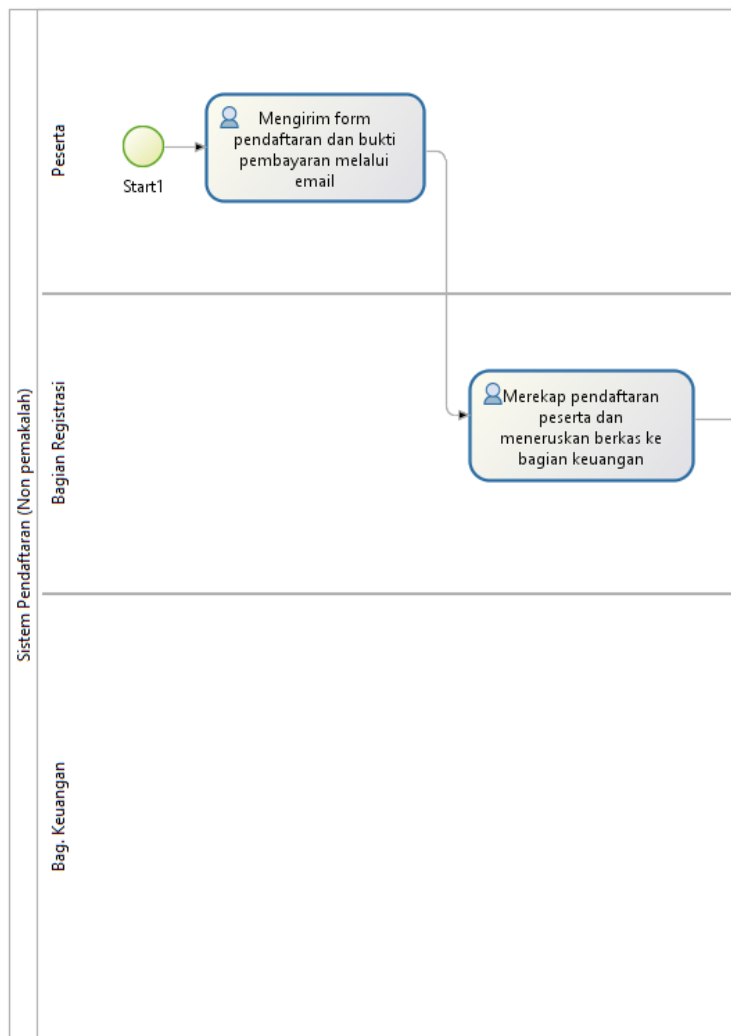
**Tabel A.2 Hasil wawancara dengan narasumber**

Pertanyaan	Jawaban
Bagaimana proses yang terjadi sekarang pada pendaftaran ISICO Bu?	Jadi di ISICO itu ada dua <i>user</i> nya, yaitu non pemakalah dan pemakalah. Untuk pemakalah, mereka harus mendaftar dengan mengisi <i>form</i> terlebih dahulu.
<i>Form</i> nya itu dimana Bu?	<i>Form</i> nya ya melalui <i>email</i> itu. Jadi di <i>form</i> tersebut isi biodata diri dan nomor ID makalahnya. Lalu <i>form</i> dikirimkan melalui <i>email</i> dan diterima oleh bagian registrasi.

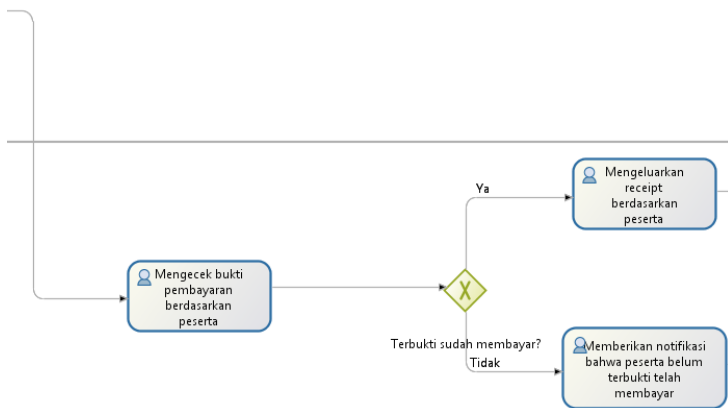
Pertanyaan	Jawaban
	<p>Bagian registrasi ini kemudian melakukan rekap dan memindahkan <i>email</i> ke bagian keuangan. Tujuannya untuk bertanya aja sih atau verifikasi apakah pembayaran yang dilakukan peserta benar atau tidak. Selain ke bagian keuangan, <i>email</i> tadi juga ditanya ke bagian <i>editor</i> juga untuk mengecek makalahnya kelebihan halaman atau tidak. Nah kalau sudah benar semuanya, nanti bagian keuangan akan mengeluarkan <i>invoice</i> terus diberikan pada bagian registrasi. Dari bagian registrasi akan dikirimkan lagi ke peserta.</p>
<p>Kalau untuk non pemakalah bagaimana Bu?</p>	<p>Prosesnya sama, cuma untuk non pemakalah pastinya tidak perlu mengisi ID makalahnya. Jadi hanya melakukan pembayaran peserta gitu saja.</p>



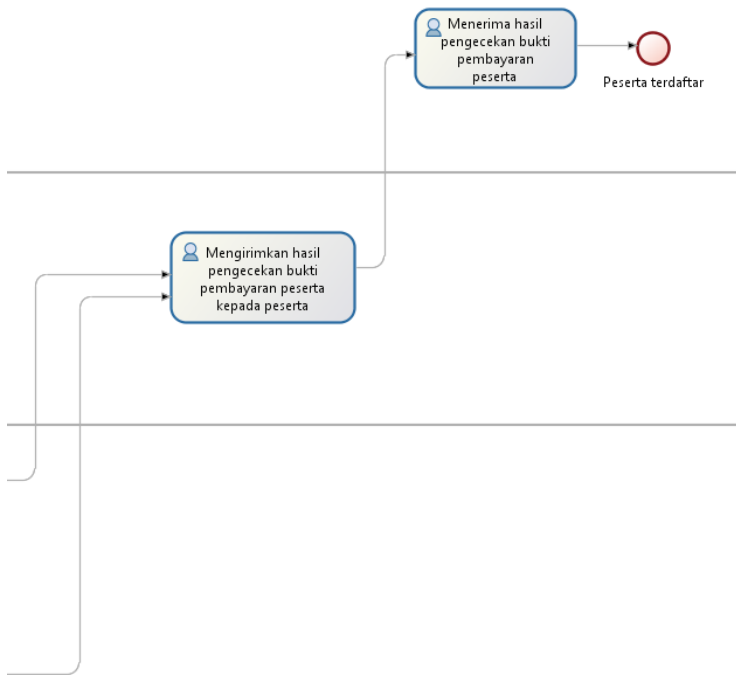
## Lampiran B. Diagram Proses (*as-is*)



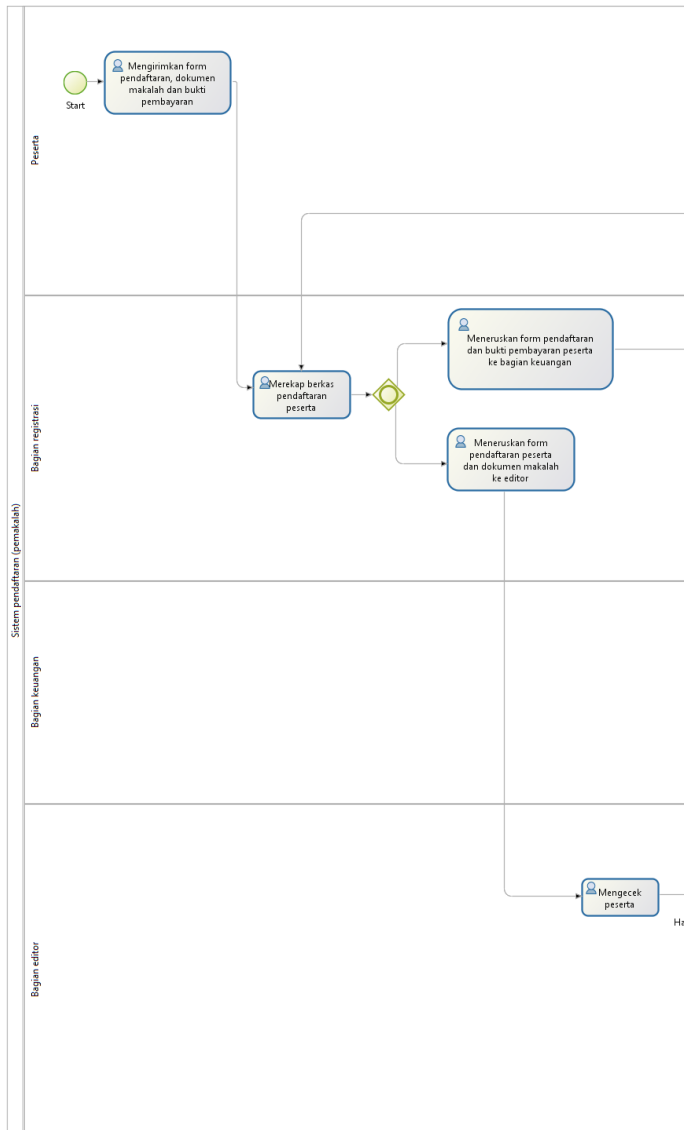
Gambar B.1 Diagram proses *as-is* non pemakalah bagian 1



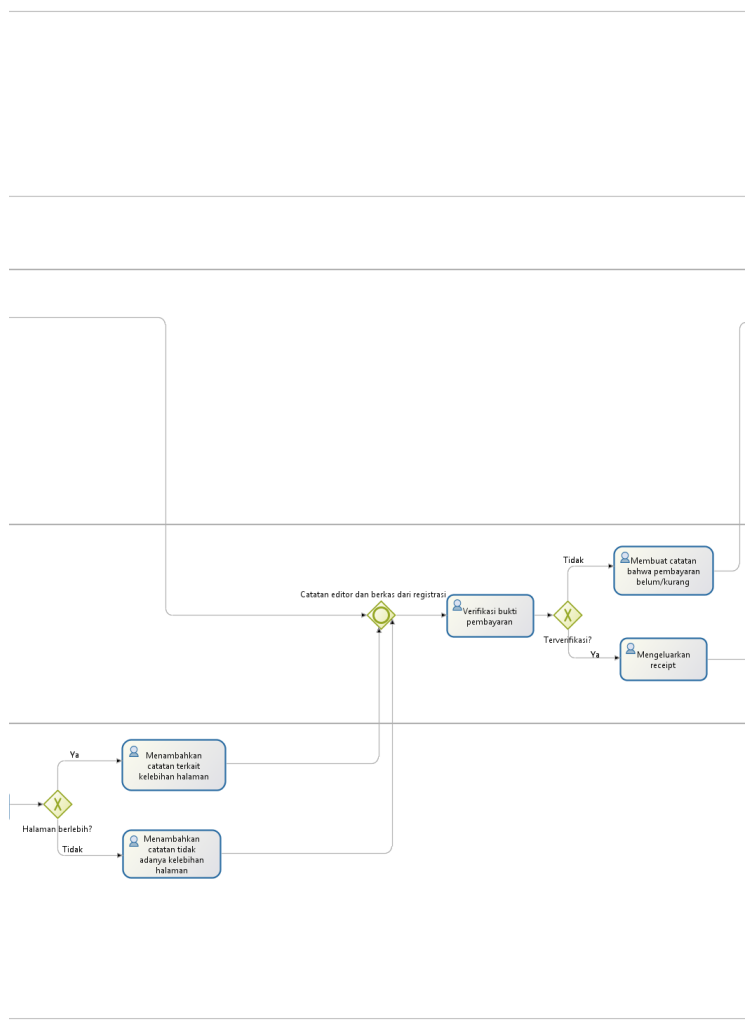
**Gambar B.2 Diagram proses *as-is* non pemakalah bagian 2**



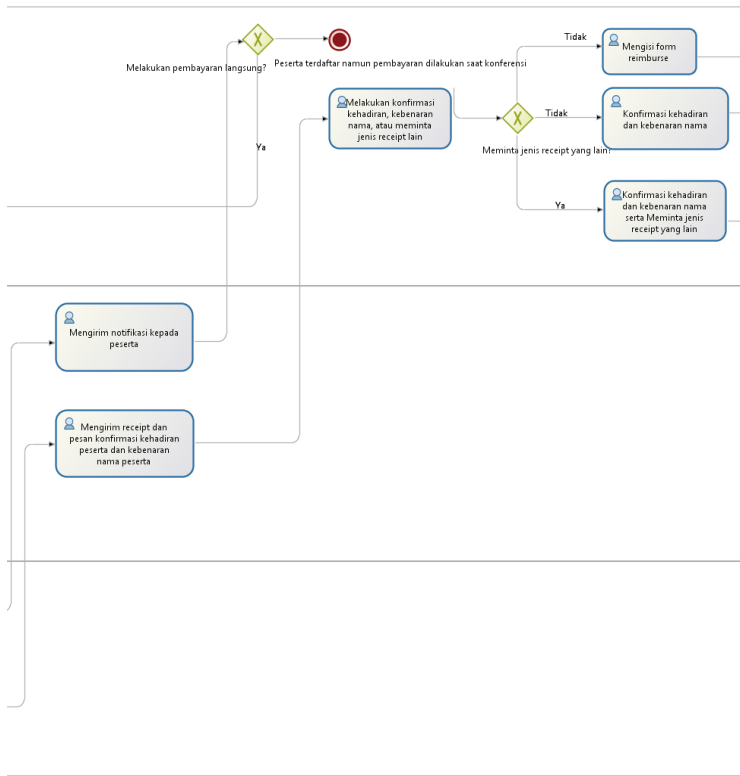
**Gambar B.3** Diagram proses *as-is* non pemakalah bagian 3



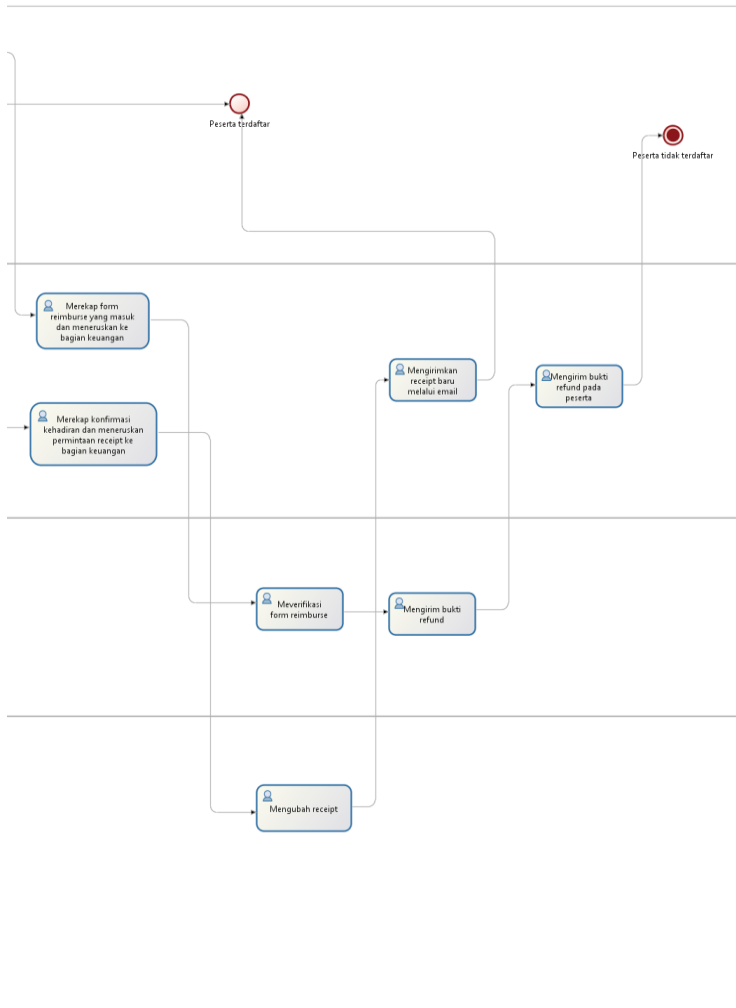
**Gambar B.4 Diagram proses as-is pemakalah bagian 1**



**Gambar B.5 Diagram proses *as-is* pemakalah bagian 2**



**Gambar B.6** Diagram proses *as-is* pemakalah bagian 3



**Gambar B.7 Diagram proses *as-is* pemakaian bagian 4**

*Halaman ini sengaja dikosongkan*



## Lampiran C. Hasil Wawancara Proses *to-be* ISICO

Tabel C.1 berisikan informasi wawancara proses *to-be* ISICO sedangkan Tabel C.2 berisi hasil wawancara dengan narasumber, yaitu penyelenggara ISICO.

**Tabel C.1 Informasi wawancara proses *to-be***

Informasi Wawancara	
<b>Narasumber</b>	Amna Shifia Nisafani
<b>Jabatan</b>	Bagian Registrasi ISICO 2017
<b>Pewawancara</b>	Delina Rahayu Effendi
<b>Jabatan</b>	Peneliti
<b>Tempat</b>	
<b>Tanggal</b>	27 September 2016
<b>Tujuan</b>	Menggali informasi mengenai proses pendaftaran yang diharapkan pada ISICO.

**Tabel C.2 Hasil wawancara proses *to-be***

Pertanyaan	Jawaban
Bagaimana proses yang diharapkan untuk ada pada sistem pendaftaran ISICO Bu?	Jadi sebenarnya adanya sistem ini itu bertujuan untuk mempermudah proses yang masih manual tadi. Oh iya, untuk lebih mempermudah ketika hari H, nanti dibuatkan <i>qr code</i> juga ya. Jadi nanti bikin <i>form</i> buat registrasi ulang yang terintegrasi dengan <i>qr code</i> sehingga saat registrasi ulang panitia tinggal <i>scan qrcodenya</i> dan otomatis peserta terdaftar menghadiri konferensi.

Pertanyaan	Jawaban
	<p>Jangan lupa juga nanti ada 2 kali registrasi buat pemakalah, yaitu registrasi kehadiran dan registrasi saat akan menyampaikan makalahnya. Nah ini semua pakai <i>qrcode</i> itu tadi. Jadi untuk pemakalah di <i>qrcodenya</i> bisa tersimpan juga informasi tentang ID pemakalahnya berapa, ruangnya berapa dan jadwalnya, serta disesuaikan dengan jumlah makalah yang didaftarkan.</p>

## Lampiran D. Diskusi Proses yang Berjalan pada Aplikasi

Tabel D.1 berisikan informasi diskusi proses yang berjalan pada aplikasi sedangkan Tabel D.2 berisi hasil diskusi terkait proses yang berjalan pada aplikasi.

**Tabel D.1 Informasi diskusi proses yang berjalan pada aplikasi**

Informasi Diskusi	
<b>Peserta Diskusi I</b>	Faizal Mahananto (FM)
<b>Jabatan</b>	Ketua ISICO 2017
<b>Peserta Diskusi II</b>	Amna Shifia Nisafani (AS)
<b>Jabatan</b>	Bagian Registrasi ISICO 2017
<b>Peserta Diskusi I</b>	Arif Wibisono (AW)
<b>Jabatan</b>	
<b>Peserta Diskusi II</b>	Delina Rahayu Effendi (DL)
<b>Jabatan</b>	Peneliti
<b>Tempat</b>	TC-101 Jurusan Sistem Informasi ITS
<b>Tanggal</b>	14 Desember 2016
<b>Tujuan</b>	Menggali informasi mengenai proses pendaftaran yang diharapkan pada ISICO.

**Tabel D.2 Hasil diskusi proses yang berjalan pada aplikasi**

Nama Narasi Diskusi	
AS	Jadi selama ini yang terjadi dalam proses pendaftaran ISICO adalah, peserta itu membayar biaya pendaftaran terlebih dahulu setelah <i>paper</i>

Nama	Narasi Diskusi
	<p>diterima lalu bagian Keuangan memberikan bukti pembayaran. Kemudian, penyelenggara memiliki <i>Google Sheet</i> yang berisi catatan ID paper pemakalah dan editor memasukkan keterangan apakah terdapat kekurangan bayar atau tidak. Dampaknya dirasakan saat registrasi hari pelaksanaan konferensi berlangsung, dimana orang yang harusnya setelah registrasi bisa langsung masuk, tetapi ini tidak. Mereka harus antri lagi di bagian keuangan untuk membayar kekurangan bayar yang disebabkan karena <i>editor</i> mengedit kekurangan bayar setelah selesai melakukan editing paper yang siap cetak, untuk menunggu kemungkinan terjadinya perubahan paper. Sehingga kalau <i>paper</i> sudah dinyatakan diterima dan peserta membayar dulu, biasanya terjadi kekurangan bayar.</p>
FM	<p>Seharusnya pembayaran <i>paper</i> itu dilakukan setelah camera ready selesai ya, supaya tidak terjadi kejadian seperti itu.</p>
AS	<p>Nah itu tergantung kebijakan Bapak. Kalau dulu, ada ketentuan bahwa <i>early bird</i> itu mulai dibuka meskipun papernya belum tentu diterima. Akhirnya kita harus mengembalikan uang kalau paper tersebut benar-benar tidak diterima.</p>
FM	<p>Saya pengennya di <i>form</i> itu nanti bisa keluar tanggalnya berapa kalau misalnya dia mau bayar paper. Jadi, misalnya kalau tanggalnya masih tanggal <i>early bird</i>, berarti dia dikasih harga yang <i>early bird</i>. Lalu yang disubmit ini adalah paper yang sudah pasti diterima gitu.</p>
AS	<p>Nah sekarang pertanyaannya adalah, <i>early bird</i>nya itu kapan? Bapak harus menentukan tanggalnya</p>

Nama	Narasi Diskusi
	terlebih dahulu. Jangan sampai jadwal <i>early bird</i> itu sebelum pengumuman <i>acceptance paper</i> karena nanti bisa muncul <i>refund</i> . Artinya, kalau sebelum pengumuman <i>acceptance</i> sudah bisa mendaftar <i>early bird</i> , pasti peserta akan meminta uangnya kembali kalau akhirnya papernya tidak diterima. Dan lagi, diterimanya itu setelah revisinya selesai atau setelah dilakukan review?
FM	Oh gitu ya. Berarti pembukaan pendaftaran yang <i>early bird</i> dilakukan setelah semua sudah <i>fix</i> saja, yaitu setelah <i>camera ready</i> .
AS	Jadi yang dibutuhkan delina adalah 1) list paper (di <i>database</i> ) 2) kelebihan <i>paper</i> 3) sama <i>email</i> . Loginnya nanti kalau bisa sama kayak <i>login easy chair</i> , yaitu dengan menggunakan alamat email yang ada di paper. Jadi nanti peserta <i>sign up</i> , registrasi berapa paper yang mau didaftarkan, terus datanya muncul id <i>paper</i> , <b>author</b> , terus pilih <i>presenter</i> nya siapa, terus habis gitu ada kelebihan <i>paper</i> . Nah kalo nanti <i>presenter</i> nya adalah orang yang sama dengan registran, maka dia terhitung biaya <i>paper</i> tambahan.
FM	Untuk jumlah <i>paper</i> yang bisa didaftarkan per registran lebih baik dibatasi 3 saja.
AS	Ok. Lalu nanti di <i>invoice</i> ada tulisan <i>invoice due date</i> (max 1 hari). Tapi, kalau ternyata dia minta perpanjangan waktu bayar gimana?
FM	Ya tidak bisa, dia harus melakukan registrasi ulang kalau gitu.
AS	Jadi sebenarnya prosesnya sederhana lho. Pertama, daftar akun, registrasi, melakukan pembayaran,

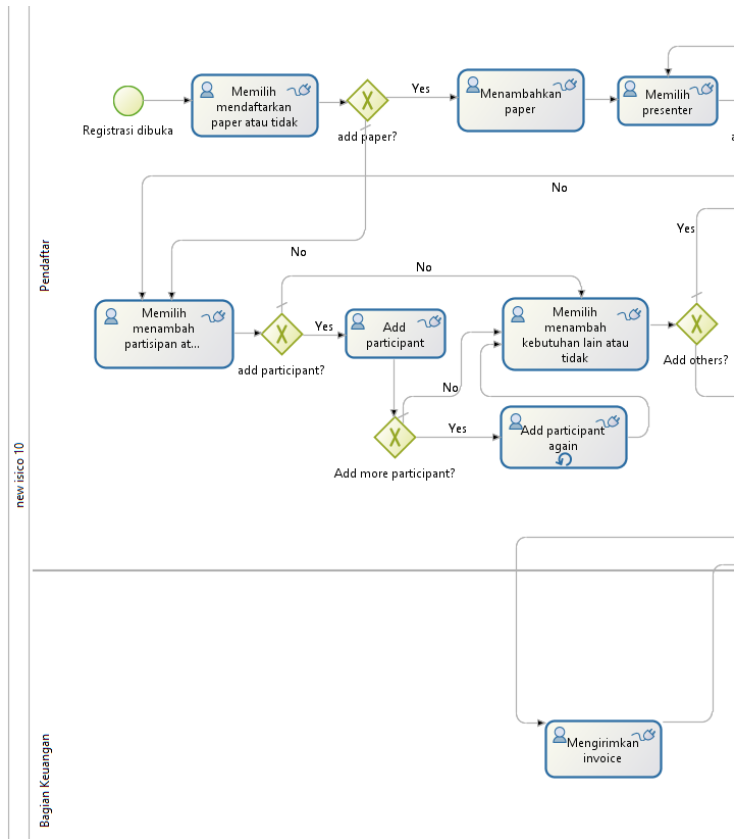
Nama	Narasi Diskusi
	terus Mas Ipung konfirmasi pembayaran.
AW	Mas Ipung bisa ga dapet <i>report</i> siapa aja yang uda bayar?
AS	Bisa
AW	Proses <i>aware</i> itu yang susah adalah report. Nanti pake phpmaker aja.
AS	Kalo uda konfirmasi pembayaran, peserta bisa cetak bukti pembayaran dan <i>qrcode</i> .
FM	<i>Qrcode</i> ini alatnya bagaimana?
AS	<i>Qrcode</i> aja lho nanti pake kamera. Jadi tinggal discan.
FM	Lah tapi kan nanti prosesnya lama, jadi harus terhubung internet dulu baru bisa registrasi.
AS	Engga, bisa kok.
AW	Ga lama kok, kalo lama pun masih reliabel.
AS	Qrcodenya per <i>presenter</i> ya. Nanti ada 3 <i>qrcode</i> atas nama nya siapa.
FM	Kalo nanti dia ada kekurangan bayar gimana?
AS	Ga mungkin ada kekurangan bayar, kan bayarnya setelah ini tadi.
FM	Jadi ini produknya, <i>paper</i> , <i>non author</i> , prosiding, hotel.
AS	Jadi <i>scopenya</i> adalah setelah dia sudah <i>camera ready</i> selesai.
AS	Tiap ada yang <i>upload</i> , Mas Ipung dikasih <i>email</i> .
FM	Di <i>qrcode</i> ada jadwal presentasi?

Nama	Narasi Diskusi
AS	Iya.
DL	Kalo audience aja gimana bu?
As	Ya sama aja. Jadi ini registrasinya satu aja, dia milih kalo ga ada <i>paper</i> ya langsung masuk ke registrasi <i>non author</i> , nanti yang didaftarkan adalah nama, instansi, terus yang lain-lain (kayak hotel dan prosiding).

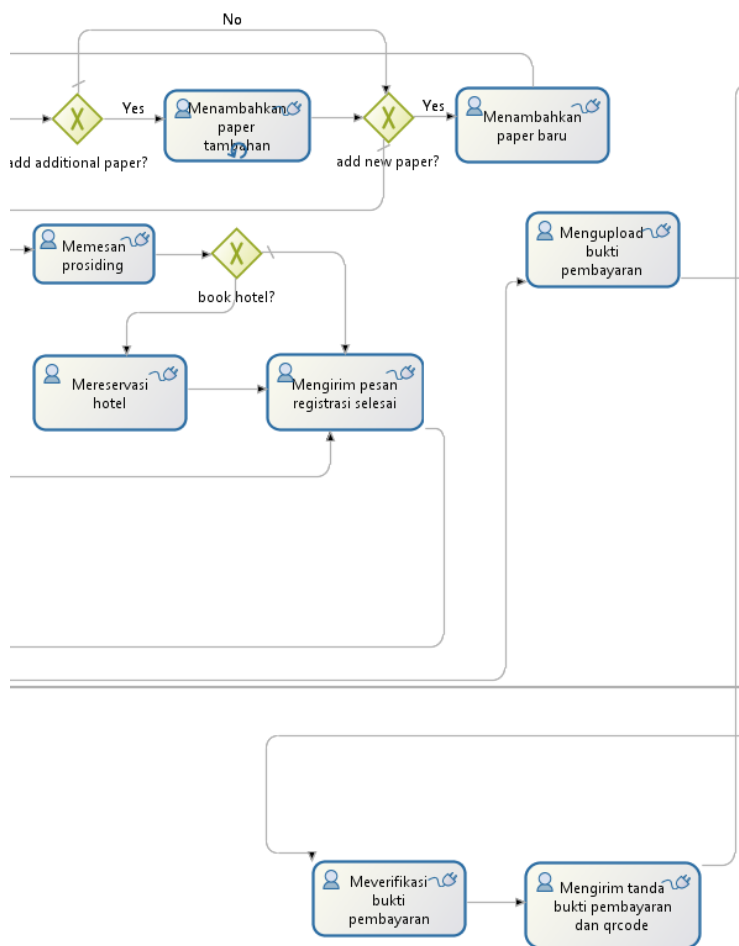
*Halaman ini sengaja dikosongkan*



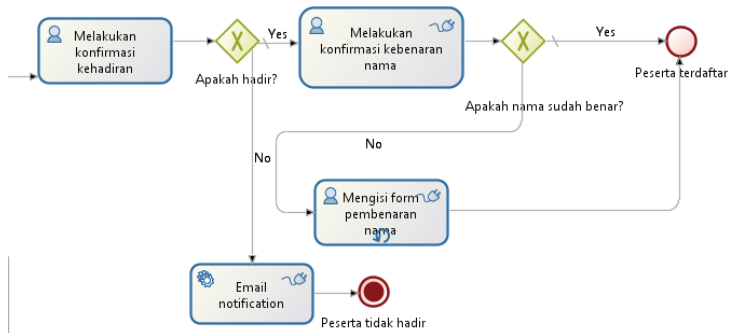
## Lampiran E. Diagram Proses (*to-be*)



Gambar E.1 Diagram proses *to-be* bagian 1



**Gambar E.2** Diagram proses *to-be* bagian 2



**Gambar E.3** Diagram proses *to-be* bagian 3

*Halaman ini sengaja dikosongkan*

## Lampiran F. Tampilan Hasil Implementasi

The screenshot displays the Bonitasoft web application interface for user registration. The top navigation bar is red with the Bonitasoft logo on the left and user information (Welcome: User) and a Settings link on the right. Below the navigation bar, there are three tabs: Tasks (selected), Cases, and Processes. The main content area is divided into two columns. The left column contains a sidebar with a back arrow, a 'Filters' section with a 'Process' dropdown set to 'All' and a search bar, and a 'Task list' section with 'TAKE' and 'RELEASE' buttons. The task list table has columns for 'Task name', 'Process name', and 'Due date', with one entry 'Pendaftaran Akun' due on 'Apr 13 4:15 PM'. The right column contains a 'Form' tab, 'Comments', and 'Overview' links. The form is titled 'Pendaftaran Akun' and includes input fields for 'Name', 'Username', 'Password', and 'Email', each with a red asterisk indicating a required field. There is an 'Add again' checkbox and a 'SEND' button. The footer of the form area shows 'Bonitasoft © 2016'.

**Bonitasoft**

Welcome: User Settings

Tasks Cases Processes

Filters

Process All Search...

Task list

TAKE RELEASE 1 - 1 / 1

Task name	Process name	Due date
Pendaftaran Akun	Pendaftaran Akun	Apr 13 4:15 PM

1 - 1 / 1

Form Comments Overview

**Pendaftaran Akun**

Name \*

Username \*

Password \*


Email \*

Add again

SEND

Bonitasoft © 2016

Gambar F.1 Mendaftarkan akun Bonita

new isico 10

marinasaf | [Logout](#) | [Bonita BPM Portal](#)

### Paper Registration

Do you want to add paper?

☐ Yes  
☐ No

[SUBMIT](#)

Bonitasoft © 2016

**Gambar F.2** Memilih mendaftarkan *paper* atau tidak

### Add Paper

Paper id \*

SUBMIT

**Gambar F.3 Menambahkan *paper***

### Set Presenter

Paper id

Paper title

Select presenter \*



Status \*



Add additional paper

☐

**Gambar F.4 Memilih *presenter***



### Add Additional Paper

Paper presenter


Remaining paper

Select Paper \*

Add other paper

☐

**Gambar F.5 Menambahkan *paper* tambahan**

new isico 10

marinasaf | [Logout](#) | [Bonita BPM Portal](#)

### Participant Registration

Do you want to add participant?

☒ Yes  
☐ No

**SUBMIT**

Bonitasoft © 2016

**Gambar F.6 Memilih menambahkan partisipan atau tidak**

### Add Participant

First name \*

Middle name

Last name \*

Institution \*

Add participant

☐

SUBMIT

**Gambar F.7 Menambahkan partisipan**

new isico 10

marinasaf | [Logout](#) | [Bonita BPM Portal](#)

### Memilih Menambah Kebutuhan Lain Atau Tidak

Do you want to add other?

☐ Yes  
☐ No

**SUBMIT**

Bonitasoft © 2016

**Gambar F.8 Memilih menambahkan kebutuhan lain atau tidak**

### Proceeding Order

Proceeding quantity


 ▼

Hotel reservation

☐

SUBMIT

**Gambar F.9 Memesan prosiding**

new isico 10

marinasaf | Logout | Bonita BPM Portal

### Book Hotel

Passport ID \*

Full name \*

Gender \*

☐ Male  
☐ Female

Country \*

Select hotel \*

Select hotel ▼

Check in \*

pick  
12

Check out \*

pick  
12

Number of room \*

Guest number \*

SUBMIT

Gambar F.10 Mereservasi hotel



**Bonitasoft** Welcome: **Marina Safitri** User Settings

Tasks Cases Processes

Form Comments Overview

### Mengupload Bukti Pembayaran

Invoice id  
201704102QITX2

Invoice  
☐ URL ☒ File  
[201704102QITX2.pdf](#)  
[modify](#) [remove](#)

Payment Receipts \*  
Browse...

SUBMIT

Bonitasoft © 2016

CLOSE

Gambar F.12 Mengunggah bukti pembayaran





The screenshot displays the Bonitasoft web application interface. At the top, there is a red navigation bar with the Bonitasoft logo on the left and user information (Welcome: User) and a Settings link on the right. Below the navigation bar, a red sidebar contains icons for Tasks, Cases, and Processes. The main content area features a modal form titled "Mengirim Tanda Bukti Pembayaran dan Qrcode". The form has three tabs: "Form" (selected), "Comments", and "Overview". It contains two input fields: "Payment Receipt \*" and "QR Code \*", each with a "Browse..." button. A "SEND" button is positioned below the input fields. At the bottom of the modal, there is a "CLOSE" button. A footer bar at the bottom of the modal displays "Bonitasoft © 2016".

**Bonitasoft**

Welcome: User | Settings

Tasks Cases Processes

Form Comments Overview

**Mengirim Tanda Bukti Pembayaran dan Qrcode**

Payment Receipt \* Browse...

QR Code \* Browse...

SEND

Bonitasoft © 2016

CLOSE

**Gambar F.14 Mengirim tanda bukti penerimaan pembayaran dan *qrcode***

Welcome: **Marina Safitri**
User
Settings

Tasks
Cases
Processes

To do
1

My tasks
1

Done tasks

Filters

Process

Search...

Task list

TAKE
RELEASE

1 - 1 / 1

<input type="checkbox"/>	Task name	Process name	Due date
<input type="checkbox"/>	Melakukan konfirmasi kehadiran	new isico 10	Apr 10 8:48 AM

1 - 1 / 1

Form
Comments
Overview

Melakukan Konfirmasi Kehadiran

Here are your qrcode and payment receipts

Payment receipts

☐ URL
☒ File

201704102QITX2 (1).pdf

modify
remove

Qrcode

☐ URL
☒ File

2.tar

modify
remove

Will you attend the conference?

☒ Yes
☐ No

SUBMIT

Gambar F.15 Konfirmasi kehadiran

185

**Bonitasoft** Welcome: **Marina Safitri** User Settings

**Tasks** Cases Processes

**Filters**

Process  Search...

In task name column

**Task list**

1 - 1 / 1

<input type="checkbox"/>	<input type="button" value="User"/> Task name <input type="button" value="Dropdown"/>	Process name	Due date
<input type="checkbox"/>	Melakukan konfirmasi kebenaran nama	new isico 10	Apr 10 8:51 AM

1 - 1 / 1


**Melakukan Konfirmasi Kebenaran Nama**



Is/are the names right?

☒ Yes ☐ No

Bonitasoft © 2016

**Gambar F.16 Konfirmasi kebenaran nama**

**Bonitasoft** Welcome: **Marina Safitri**  **User** | [Settings](#)

☒ **Tasks**  **Cases**  **Processes**

[Form](#) [Comments](#) [Overview](#)

### Mengisi Form Pembetulan Nama

ID \*

First name \*

Middle name

Last name \*

Type \*

**CLOSE**

**Gambar F.17 Form pembetulan nama**

*Halaman ini sengaja dikosongkan*

## Lampiran G. Hasil User Acceptance Testing

Tabel G.1 sampai dengan Tabel G.4 merupakan hasil pengujian pertama yang dilakukan. Namun ternyata masih terdapat kesalahan pada saat pembuatan *invoice* ketika peserta hanya mendaftarkan *paper* saja tanpa mendaftarkan partisipan dimana tertulis dalam menjalankan *test script* kedua. Akibatnya proses setelah pembuatan *invoice*, dimana pada Bonita yaitu sebelum mengirimkan *invoice*, hingga akhir jadi tidak dapat dilanjutkan.

**Tabel G.1 Checklist item keberhasilan aktivitas pada proses**

No.	Aktivitas	Pass	Fail
	<b>Peserta</b>		
1	Memilih mendaftarkan <i>paper</i> atau tidak	V	
2	Menambahkan <i>paper</i>	V	
3	Memilih <i>presenter</i>	V	
4	Menambahkan <i>paper</i> tambahan	V	
5	Memilih menambahkan partisipan atau tidak	V	
6	Menambahkan partisipan	V	
7	Memilih menambahkan kebutuhan lain atau tidak	V	
8	Memesan prosiding	V	
9	Mereservasi hotel	V	
10	Mengunggah bukti pembayaran	V	
11	Melakukan konfirmasi kehadiran	V	

No.	Aktivitas	Pass	Fail
12	Melakukan konfirmasi kebenaran nama	V	
13	Mengisi <i>form</i> pembenaran nama	V	
	<b>Bagian Keuangan</b>		
14	Mengirimkan <i>invoice</i>	V	
15	Meverifikasi bukti pembayaran	V	
16	Mengirimkan <i>qrcode</i> dan bukti penerimaan pembayaran	V	

Keterangan:

*Pass* : Aktivitas tidak dapat berjalan

*Fail* : Aktivitas dapat berjalan

**Tabel G.2 Checklist item keberhasilan notifikasi email**

No.	Pemberitahuan	Pass	Fail
1	Peserta baru kepada peserta	V	
2	Peserta baru kepada bagian keuangan	V	
3	<i>Invoice</i> terkirim kepada peserta	V	
4	Bukti pembayaran terkirim kepada bagian keuangan	V	
5	Pembayaran terkonfirmasi kepada peserta	V	
6	<i>Qrcode</i> dan bukti penerimaan pembayaran terkirim kepada peserta	V	



No.	Pemberitahuan	Pass	Fail
7	Pesan ketidakhadiran kepada peserta	V	
8	Proses pendaftaran ISICO selesai kepada peserta	V	

Keterangan:

*Pass* : Pemberitahuan berjalan dengan normal

*Fail* : Pemberitahuan tidak dapat berjalan

**Tabel G.3 Checklist item keberhasilan fungsional database**

No.	Fungsi	Pass	Fail
1	Memperbarui <i>paper</i> yang terdaftar berdasarkan ID <i>paper</i>	V	
2	Mengambil <i>membership presenter</i>	V	
3	Mengambil <i>presenter</i> berdasarkan ID <i>paper</i> yang terdaftar	V	
4	Mengambil judul <i>paper</i> berdasarkan ID <i>paper</i> yang terdaftar	V	
5	Memperbarui <i>presenter paper</i> yang dipilih	V	
6	Mengambil jumlah sisa <i>paper</i> pada suatu <i>paper</i> (saat peserta melakukan pendaftaran <i>paper</i> tambahan)	V	
7	Mengambil judul <i>paper</i> sesuai dengan <i>presenter</i> yang terpilih (saat peserta melakukan pendaftaran <i>paper</i> tambahan)	V	
8	Memperbarui <i>presenter paper</i>	V	

No.	Fungsi	Pass	Fail
	tambahan yang didaftarkan		
9	Menyimpan data partisipan	V	
10	Menyimpan data pemesanan prosiding	V	
11	Mengambil data hotel untuk layanan reservasi hotel	V	
12	Menyimpan data reservasi hotel	V	
13	Memperbarui bahwa <i>invoice</i> telah lunas	V	
14	Memperbarui bahwa <i>paper</i> yang tertulis pada <i>invoice</i> telah lunas	V	
15	Memperbarui bahwa partisipan yang tertulis pada <i>invoice</i> telah lunas	V	
16	Menyimpan data pembenaran nama	V	

Keterangan:

*Pass* : Fungsi berjalan dengan normal

*Fail* : Fungsi tidak dapat berjalan

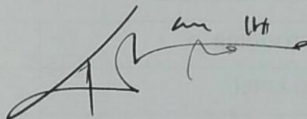
**Tabel G.4 Checklist item fungsional sistem eksternal**

No.	Aktivitas	Pass	Fail
1	Pendaftaran akun Bonita	V	
2	Pembuatan <i>invoice</i>		V
3	Pembuatan bukti penerimaan pembayaran	V	

No.	Aktivitas	Pass	Fail
4	Pembuatan <i>qrcode</i>	V	
5	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang terdaftar	V	
6	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang telah hadir	V	
7	<i>Dashboard</i> informasi jumlah partisipan yang terdaftar	V	
8	<i>Dashboard</i> informasi jumlah partisipan yang telah hadir	V	
9	<i>Dashboard</i> informasi jumlah <i>invoice</i> yang telah lunas	V	

Surabaya, 21 April 2017

Menyetujui,



Faizal Mahananto, S.Kom., M.Eng.

Tabel G.5 sampai dengan Tabel G.8 merupakan hasil pengujian kedua yang dilakukan. Di sini semua aktivitas dan fungsional telah berjalan dengan baik.

**Tabel G.5 Checklist item keberhasilan aktivitas pada proses**

No.	Aktivitas	Pass	Fail
	<b>Peserta</b>		
1	Memilih mendaftarkan <i>paper</i> atau tidak	V	
2	Menambahkan <i>paper</i>	V	
3	Memilih <i>presenter</i>	V	
4	Menambahkan <i>paper</i> tambahan	V	
5	Memilih menambahkan partisipan atau tidak	V	
6	Menambahkan partisipan	V	
7	Memilih menambahkan kebutuhan lain atau tidak	V	
8	Memesan prosiding	V	
9	Mereservasi hotel	V	
10	Mengunggah bukti pembayaran	V	
11	Melakukan konfirmasi kehadiran	V	
12	Melakukan konfirmasi kebenaran nama	V	
13	Mengisi <i>form</i> membenaran nama	V	
	<b>Bagian Keuangan</b>		
14	Mengirimkan <i>invoice</i>	V	

No.	Aktivitas	Pass	Fail
15	Meverifikasi bukti pembayaran	V	
16	Mengirimkan <i>qr code</i> dan bukti penerimaan pembayaran	V	

Keterangan:

*Pass* : Aktivitas tidak dapat berjalan

*Fail* : Aktivitas dapat berjalan

**Tabel G.6 Checklist item keberhasilan notifikasi email**

No.	Pemberitahuan	Pass	Fail
1	Peserta baru kepada peserta	V	
2	Peserta baru kepada bagian keuangan	V	
3	<i>Invoice</i> terkirim kepada peserta	V	
4	Bukti pembayaran terkirim kepada bagian keuangan	V	
5	Pembayaran terkonfirmasi kepada peserta	V	
6	<i>Qrcode</i> dan bukti penerimaan pembayaran terkirim kepada peserta	V	
7	Pesan ketidakhadiran kepada peserta	V	
8	Proses pendaftaran ISICO selesai kepada peserta	V	

Keterangan:

*Pass* : Pemberitahuan berjalan dengan normal

*Fail* : Pemberitahuan tidak dapat berjalan

Tabel G.7 Checklist item keberhasilan fungsional database

No.	Fungsi	Pass	Fail
1	Memperbarui <i>paper</i> yang terdaftar berdasarkan ID <i>paper</i>	V	
2	Mengambil <i>membership presenter</i>	V	
3	Mengambil <i>presenter</i> berdasarkan ID <i>paper</i> yang terdaftar	V	
4	Mengambil judul <i>paper</i> berdasarkan ID <i>paper</i> yang terdaftar	V	
5	Memperbarui <i>presenter paper</i> yang dipilih	V	
6	Mengambil jumlah sisa <i>paper</i> pada suatu <i>paper</i> (saat peserta melakukan pendaftaran <i>paper</i> tambahan)	V	
7	Mengambil judul <i>paper</i> sesuai dengan <i>presenter</i> yang terpilih (saat peserta melakukan pendaftaran <i>paper</i> tambahan)	V	
8	Memperbarui <i>presenter paper</i> tambahan yang didaftarkan	V	
9	Menyimpan data partisipan	V	
10	Menyimpan data pemesanan prosiding	V	
11	Mengambil data hotel untuk layanan reservasi hotel	V	
12	Menyimpan data reservasi hotel	V	

No.	Fungsi	Pass	Fail
13	Memperbarui bahwa <i>invoice</i> telah lunas	V	
14	Memperbarui bahwa <i>paper</i> yang tertulis pada <i>invoice</i> telah lunas	V	
15	Memperbarui bahwa partisipan yang tertulis pada <i>invoice</i> telah lunas	V	
16	Menyimpan data membenaran nama	V	

Keterangan:

*Pass* : Fungsi berjalan dengan normal

*Fail* : Fungsi tidak dapat berjalan

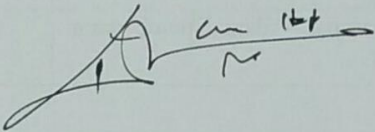
**Tabel G.8 Checklist item fungsional sistem eksternal**

No.	Aktivitas	Pass	Fail
1	Pendaftaran akun Bonita	V	
2	Pembuatan <i>invoice</i>	V	
3	Pembuatan bukti penerimaan pembayaran	V	
4	Pembuatan <i>qrcode</i>	V	
5	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang terdaftar	V	
6	<i>Dashboard</i> informasi jumlah <i>author/presenter</i> yang telah hadir	V	
7	<i>Dashboard</i> informasi jumlah partisipan yang terdaftar	V	
8	<i>Dashboard</i> informasi jumlah partisipan yang telah hadir	V	

No.	Aktivitas	Pass	Fail
9	<i>Dashboard</i> informasi jumlah <i>invoice</i> yang telah lunas	V	

Surabaya, 15 Mei 2017

Menyetujui,



Faizal Mahananto, S.Kom., M.Eng.



## Biodata Penulis



Penulis lahir di Surabaya pada 17 Juli 1995. Penulis merupakan anak ketiga dari Bapak Jeung Dody Effendi dan Ibu Suti'ah. Penulis telah menempuh pendidikan formal di SD Mardi Sunu Surabaya, SMP Negeri 4 Surabaya, dan SMA Negeri 9 Surabaya. Pada 2013, penulis diterima di Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember (ITS) Surabaya melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) dan terdaftar dengan NRP 5213100106. Selain pada bidang teknologi informasi yang ditekuni pada jurusan ini, penulis memiliki ketertarikan dalam bidang jurnalistik dan menulis. Selama masa kuliah, penulis memiliki beberapa pengalaman organisasi di ITS, diantaranya adalah menjadi anggota Departemen Media Informasi, Himpunan Mahasiswa Sistem Informasi (HMSI), sekretaris ITS TV masa jabatan 2015-2016, dan wakil ketua ITS TV masa jabatan 2016-2017. Selain itu, penulis juga sering mengikuti kepanitiaan, seperti ketua pelaksana Pelatihan Jurnalistik Tingkat Dasar yang diadakan HMSI, anggota tim publikasi dan dokumentasi ISE 2014, tim konseptor ISE 2015, serta pernah tergabung menjadi *volunteer* Pejuang Tumbler ITS. Penulis juga pernah melakukan kerja praktik di PT. Pertamina EP pada Divisi ICT Solution, Fungsi Business Solution selama 2 bulan pada tahun 2016.

Dalam pengerjaan tugas akhir di Jurusan Sistem Informasi ITS, penulis mengambil bidang minat Sistem Enterprise dengan topik Business Process Management. Apabila terdapat kritik, saran, atau pertanyaan mengenai tugas akhir ini, dapat dikirimkan pada *email* penulis: [delinare.its@gmail.com](mailto:delinare.its@gmail.com).